Abstract—This paper presents a method to recover a spatially varying illuminant colour estimate from scenes lit by multiple light sources. Starting with the image formation process, we formulate the illuminant recovery problem in a statistically data-driven setting. To do this, we use a factor graph defined across the scale space of the input image. In the graph, we utilise a set of illuminant prototypes computed using a data driven approach. As a result, our method delivers a pixelwise illuminant colour estimate being devoid of libraries or user input. The use of a factor graph also allows for the illuminant estimates to be recovered making use of a maximum a posteriori (MAP) inference process. Moreover, we compute the probability marginals by performing a Delaunay triangulation on our factor graph. We illustrate the utility of our method for pixelwise illuminant colour recovery on widely available datasets and compare against a number of alternatives. We also show sample colour correction results on real-world images.

Index Terms—Ensembled pixelwise illuminant estimation, factor graphs, scale space, colour constancy.

I. INTRODUCTION

The apparent colour of an object depends on a number of factors. One of these is the power spectrum of the lights illuminating the scene. As a result, the recovery of the illuminant colour as a means to photometric invariance has found applications in areas such as object recognition [1], [2], visual surveillance [3], white balancing [4], digital media production [5] and visual tracking [6].

Despite its importance, the recovery and identification of the illuminant colour in a 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 pixelwise illuminant from a single image is an under-constrained problem [7], [8]. As a result, existing methods often assume uniform illumination across the scene [9], [7], [10]. Furthermore, additional constraints or assumptions are used to render the problem tractable. For instance, the grey-world method [9] renders the mean reflectance across the scene to be equivalent to the illuminant colour. The grey-edge [11] adopts a similar assumption, whereby the mean reflectance difference computed from the image colour derivatives is also assumed to be identical to uniform illuminant colour. The white patch algorithm [12] recovers the illuminant colour which is correlated to the maximum values of each colour band. The shades of grey method [13] builds upon the grey-world and white patch algorithms and assumes the Minkowski norm of the derivatives of the reflectance of objects in the scene is achromatic.

Some of these approaches [7], [14], [15] also employ Bayesian statistics. This is because statistical inference allows for priors and uncertainty to be used for the illuminant recovery process. The classical Bayesian colour constancy method in [7] models the reflectance distribution as a mixture of Gaussians, where the illuminant is recovered using the posterior computed via a minimum risk rule. Rosenberg et al. [14] build upon the approach in [7] and employ a histogram to estimate the reflectance distributions using a clipping function. Gehler et al. [15] introduce a parameter to control the effect of the bin clipping function in [14].

Despite being effective, the assumption of a uniform illuminant colour as applied by the methods above implies that, in practice, these approaches can only tackle scenes lit by a single illuminant [8]. This contrasts with real world images, which often depict scenes lit by multiple lights sources. We illustrate this in Figure 1, where we show a scene lit by multiple illuminants. Note that, the original image, on the left-hand panel, has several round spot lamps illuminating the paintings on the wall. In the right-hand panel, we show the image where the effects of the illuminant colours have been removed. Note that the background wall no longer exhibits the “warm” tone induced by the lights.

To handle varying illumination, methods such as the retinex algorithm [12], [16] and Xiong et al. [17] assume the illumination varies smoothly across the scene. This is based upon the notion that the degree of change in independent colour channels can be employed to determine whether these variations correspond to variations in the illumination colour, or the reflectance of the objects in the scene. Other methods segment the image into regions of constant illumination or perform colour constancy without recovering the illuminant colour explicitly. For instance, Wang and Samaras [18] detect
and estimate the illuminants in the scene by making use of a recursive least squares method to segment the surfaces in the scene into separate light patches. Ebner [10], on the other hand, employs the local average color to perform color constancy irrespective of the illuminants used in the scene.

To account for illumination variations in the scene, thresholds on the derivative of the logarithm of color channels have been employed so as to impose a smoothness constraint on the scene irradiance. In [19], Barnard et al. used smoothness constraints on both the reflectance and illumination gamuts to identify varying illumination. The method in [19] employs the white patch algorithm [12] to account for skewed image brightness. In fact, the use of single-illuminant methods for multiple-illuminant recovery is not unusual. For instance, Gijsenij et al. [4] solve the white balance problem by restricting the scene to two known illuminant colours and estimating the illuminant variations of the illuminant estimate as there were choices delivered by existing color constancy methods. These initial estimates are also employed to recover the illuminant prototypes. The pixelwise illuminant can hence be viewed as a weighted linear combination of these set of prototype illuminants. This linear combination is such that the weight of an illuminant prototype at a given pixel corresponds to an a posteriori probability which is derived from our factor graph. Moreover, by construction, the pixelwise illumination at each scale of a subgraph is also a weighted linear combination of the illuminants across the subgraphs comprising our factor graph. Thus, the pixelwise illuminant can be viewed as the geometric mean of the illuminants across all subgraphs. In this manner the pixelwise illumination is the result of a statistical inference process across all subgraphs. Note that our method is data driven, whereby the prototype illuminants are determined statistically from the image itself without the need for additional constraints on the reflectance or illumination across the scene.

It is worth noting that, despite the approach presented here being somewhat reminiscent to that in [29], it has a number of differences with respect to the work elsewhere in literature. Recall that the methods in [29], [28], [21] employ, as an integral part of the pixelwise illuminant estimate, the initialisation delivered by colour constancy methods such as the white patch [12] or shades of gray [13]. As a result, the output of these methods depends upon the chosen initialisation approach. This, is important since there are, potentially, as many possible variants of the illuminant colour estimate as there were choices of colour constancy methods at initialisation. In contrast, the method presented here delivers a single colour illuminant estimate regardless of the number of colour constancy methods used for initialisation. Further, by making use of a factor graph and multiscale information, our approach can benefit from the strengths of common, fast colour constancy methods at initialisation as well as concurrently counter noise corruption and initialisation errors.

Furthermore, our method avoids using a predefined collection of illuminants and, instead, we opt for a grid-based approach effected by sub-dividing the image. This can be done in a number of ways. For instance, lattices [20], superpixel segmentations based on chromaticity [8], [22] or prototype illuminants [21] may be used to segment the image into uniform illuminant regions. Here we have used a uniform grid, as opposed to uniform colour clustering. The motivation for this hinges on the fact that the lattice-like subdivision of the image delivers regions where single-illuminant algorithm assumptions are more likely to hold [20]. Moreover, our method has very few free parameters, solely requiring the choice of bandwidth for the M-estimators used for the likelihood computation.

The rest of the paper is organised as follows. Section II commences by presenting the notation and background for-
malism used throughout the paper. It also presents the factor graph structure used in this paper and the inference process used to compute the pixelwise illuminant colour estimate. In Section III, we elaborate upon the practical implementation of our method, its initialisation, and the computational issues regarding the illuminant estimate recovery. In Section IV, we present a quantitative and qualitative analysis of our method and compare our results with those yielded by a number of alternatives elsewhere in literature. We also show results on the colour correction of real-world scenes. Finally, we conclude on the developments presented here in section V.

II. ILLUMINANT ESTIMATION

A. Background

As mentioned earlier, we start with the image formation process to develop our pixelwise illuminant recovery approach. This allows for an appropriate motivation and better understanding of our factor graph, its subgraphs and strata, as spanned by the image scale space and initial estimates delivered by existing colour constancy methods. Thus, in this section, we will provide some background and notation used throughout the paper regarding the scale space used for our factor graph, the image irradiance and the manner in which the pixelwise light colour estimates can be expressed as a linear weighted combination of illuminant prototypes.

To commence, recall that our factor graph is divided into subgraphs, each of which corresponds to a particular initialisation. The strata of each subgraph correspond to a different scale across the image. We have done this so as to account for the variations of the illuminant at different resolutions in the image. It is worth noting in passing that this treatment is consistent with the behaviour of the human visual system whereby information about the scene at equally distinct resolutions is used to sense different structural aspects in the image relative to its surroundings, a concept known as lateral inhibition [32]. Also, note that the notion of scale has been used in edge based colour constancy [11], where scale is viewed as the spread of a Gaussian distribution. Furthermore, the method in [11] estimates a uniform colour illuminant using the derivative of the image at a given scale. Here, we obtain different scales by smoothing the image data using the Gaussian kernel

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

where the variance \( \sigma \) represents the scale.

Figure 2 shows the effect of the scale \( \sigma \) in the Gaussian kernel in Equation (1) upon the illuminant recovered by the method of Gijsenij et al. [20]. The figure shows a wall lit by two illuminants of different colour. Note that, on the right-hand panel, the area covered by illuminants “widens” when the image is blurred. This can be noticed as the area of grey-coloured light spreads out around the lamp towards the reflection on the wall. Likewise, in the bottom-left corner, the orange hue overlaps onto the multi-coloured scene details, which are much more clearly visible at the finer scale in the left-hand panel.

For each scale, consider the spectral radiance \( I_\sigma(u, \lambda) \) at pixel \( u \) and wavelength \( \lambda \) of a scene lit by a pixelwise illuminant with power spectrum \( L_\sigma(u, \lambda) \). With this notation, we can express the scene radiance as follows

$$I_\sigma(u, \lambda) = L_\sigma(u, \lambda)R_\sigma(u, \lambda), \quad (2)$$

where \( R_\sigma(u, \lambda) \) is a function of the mean scattered power, the surface spectral reflectance and the camera spectral sensitivity function [33]. Note that \( R_\sigma(\cdot) \) above accounts for the proportion of the incident light that is reflected by the object at a particular scale \( \sigma \). This function hence defines the colour of the object at the scale under consideration. Further, the relation above has been widely used in colour constancy [34] and is consistent with the dichromatic model [35].

As mentioned earlier, each of the subgraphs in our factor graph is initialised using a set of prototypes delivered by an existing method. This is done using a set of prototypes \( \Upsilon_k \) drawn from the image itself when processed by the \( k^{th} \) existing approach under consideration, i.e. white patch [12], shades of gray [13], etc. Making use of these prototypes, we can express the illuminant at a pixel location \( u \), and at a subgraph initialisation with method \( k \) as follows

$$L_{\sigma,k}(u) = \sum_{i \in \Upsilon_k} \omega_i \cdot \ell_i, \quad (3)$$

where \( \omega_i \) is the weight of the \( i^{th} \) prototype illuminant \( \ell_i \in \Upsilon_k \). Also, from now on and throughout the paper, for the sake of clarity and without loss of generality we have omitted the wavelength variable \( \lambda \) and the subscript \( k \) on the weights \( \omega_i \) and the prototype \( \ell_i \).

The expression in Equation (3) implies that the illuminant at a pixel depends on the contributions of the data-driven illuminant prototypes \( \ell_i \) to the power spectrum of \( L_{\sigma,k}(u) \). It is worth noting that such a constrain upon the pixelwise illuminant estimates is not overly restrictive. This is because the prototypes in \( \Upsilon_k \) are pre-computed from the input image, as opposed to being chosen from a library or user input. This has the advantage that these prototypes are, by definition, drawn from the image itself and, therefore, are expected to

Fig. 2. The effect of scale on the illuminant colour. Top row: Input image at a fine (left-hand panel) and course (right-hand panel) scales, respectively; Bottom row: Illuminant colour map yielded by the method in [20] for the images in the top row.
be in good accordance with the input data. Here, we have computed these prototypes making use of the mode seeking method in [36].

B. Factor graphs

With the notation and background above, we now proceed to consider the weight \( \omega_{i,\sigma}(u) \) in Equation (3) as a probability and lay out the illuminant recovery problem in a manner akin to maximum a posteriori estimation using a factor graph [30], [31].

Recall that a factor graph \( G = (\mathcal{N}, \mathcal{E}) \) is an undirected graphical model comprised of a node set \( \mathcal{N} \) and an edge set \( \mathcal{E} \). Its node set is subdivided into two distinguishable subsets. The first of these is given by the variables \( \mathcal{V} \) whereas the other one contains the factors \( \mathcal{F} \), i.e. \( \mathcal{N} = \mathcal{V} \cup \mathcal{F} \). The edge set connects variables to factors so as to parameterise the conditional probabilities of the cliques that constitute the graph\(^1\).

As mentioned earlier, our factor graph is comprised by a set of subgraphs with image scales as strata. Figure 3 depicts the factor graph arising from the relations between the illuminant variables and the image scale space. In the left-hand panel, we show the interaction between variables initialised by the prototypes arising from different illuminant recovery methods as subgraphs, whereby each of these bears a pyramid-like structure induced by the multiple scales used over the image lattice. Note that there is a one-to-one relationship between the variables across adjacent subgraphs, while the connection from finer to coarser scales is many-to-one. Additionally, the right-hand panel of Figure 3 shows a close up of one of the subgraphs in the left-hand panel. In the figure, the factors are represented by square tokens whereas the variables are given by the circular markers along the image lattice across different scales.

In this paper, we view 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 viewed 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 [39], [40], these potentials are multivariate functions of Fourier transforms. Here, the factorisations are probabilities governed by the image data and the illuminant prototype set \( \Upsilon_k \).

Moreover, from a probabilistic standpoint, we can view the interrelated variables of the graphical structure in the right-hand side of Figure 3 as conditional distributions within and across scales of each subgraph. In this setting, if \( G_k \) denotes a factor subgraph for the \( k^{th} \) colour constancy method employed at initialisation, the complete graph \( G \) can be fully written as the union over the set of subgraphs \( \mathcal{K} \) given by

\[
G = \bigcup_{G_k \in \mathcal{K}} G_k. \tag{4}
\]

\(^1\)For an introductory treatment of factor graphs we would like to refer the interested reader to the textbooks of Bishop [37] and Koller and Friedman [38]. More information on the sum-product algorithm and its application to factor graphs can be found in [30], [31].

This is important since the probability of the pixelwise illuminants across the image \( L \) given the graph \( G \) 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

\[
P(L|G) = \frac{1}{Z} \prod_{c \in C} \psi(L_c; \bigcup_{G_k \in \mathcal{K}} G_k) = \frac{1}{Z} \prod_{G_k \in \mathcal{K}} \prod_{u,v \in G_k} P(I(u),I(v)|L_k(u),L_k(v)) \prod_{\sigma \in \Gamma, \ell_i \in \Upsilon_k} P(L_{\sigma,k}(u)|\ell_i), \tag{5}
\]

where \( \Gamma \) 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, \( \psi(L_c; G_k) \) is the potential function over connected variables in the clique \( c \in C \) and \( Z \) is a partition function.

The nature of Equation (5) is such that it effectively conditions the illuminant distribution across the image to the potential functions for the variable nodes over the graph and the prototypes in \( \Upsilon_k \). This is because each of the potential functions is defined over a subset of variables, i.e. the pixelwise illuminants \( L_c \) for the clique \( c \in C \). This has the advantage that it ensures every variable in a clique is connected to at least one other variable through a factor node. At the factor node, the conditional probabilities associated with the corresponding variables are defined by taking the sets of variables, factors and the edges that connect them together across the factor graph \( G \).

Its also worth noting that the expression in Equation (5) is motivated by Bayes’ rule. This reflects the notion that the pixelwise illuminant recovery problem can be stated in terms of the likelihood of pixel and prior distributions with the partition function \( Z \) acting as a normalisation constant. Here, we assume that the illuminant priors \( P(L_{\sigma,k}(u)|\ell_i) \) for our prototype illuminants across image scales \( \sigma \in \Gamma \) and subgraphs \( G_k \in \mathcal{K} \) are independent from each other. On the other hand, adjacent pixels on each scale relate to factor potentials which, in turn, define the likelihood function for the illuminant estimation process. The likelihood function \( P(I(u),I(v)|L_k(u),L_k(v)) \) captures the most likely state of the observed image data at neighbouring pixel sites given their respective illuminant estimates. Additionally, the priors \( P(L_{\sigma,k}(u)|\ell_i) \) correspond to the likelihood of an illuminant for a given pixel \( u \) at scale \( \sigma \) to assume the illuminant prototype \( \ell_i \) from the prototype set \( \Upsilon_k \).

It is worth mentioning that the relationship between subgraphs can be captured using a 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 (5). This is as it lends itself, by construction, to
Fig. 3. Left-hand panel: Diagram of the factor graph used throughout the paper showing the relationships between subgraphs and pixels at different scales; Right-hand panel: Detail of one of the subgraphs in the left-hand panel, where the variables are shown as round markers whereas the factors are depicted as squared tokens.

a 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

$$\arg\max_L P(L|G_k) = \arg\max_L \left\{ \prod_{G_k \in K} \prod_{u,v \in G_k} P(I(u), I(v)|L_k(u), L_k(v)) \right\} \prod_{\sigma \in \Gamma} \prod_{\ell_i \in \Upsilon_k} P(L_{\sigma,k}(u)|\ell_i)$$

(6)

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

C. Priors and Probabilities

The prior conditional probability of the target function in Equation (6) makes use of the cosine of the Euclidean angle between the pixelwise illuminant and each of the prototypes in $\Upsilon_k$. This is, effectively, a similarity measure between an illuminant at the pixel $u$ and the $i^{th}$ prototype illuminant $\ell_i$ given by

$$\cos \theta = \alpha_{u,i} = \frac{\langle L_{\sigma,k}(u), \ell_i \rangle}{\| L_{\sigma,k}(u) \| \| \ell_i \|}.$$  

(7)

where $\theta$ is the angle between the two vectors, $\langle \cdot, \cdot \rangle$ is the inner product and $\| \cdot \|$ is the vector L2 norm.

For the illuminant prior, we ensure the cosine above maps to a probability by setting

$$P(L_{\sigma}(u)|\ell_i) = \frac{1}{Y} \exp \left( - (1 - \alpha_{u,i}) \right),$$

(8)

where $Y$ is a normalisation constant given by

$$Y = \sum_{\ell_i \in \Upsilon_k} \exp \left( - (1 - \alpha_{u,i}) \right).$$

(9)

For the likelihood in Equation (6), we adopt the notion that neighbouring pixels are likely to have similar illuminant colour if their reflectance and shape are close to one another. Recall that the variable $R_{\sigma}(u, \lambda)$ in Equation (2) can be expressed in terms of the image irradiance and the illuminant power spectrum at pixel $u$.

As a result, the likelihood in Equation (6) can be written as follows

$$P(I(u), I(v)|L_k(u), L_k(v)) = \frac{1}{\Omega} K(L_k(u), L_k(v)) \times \exp \left( - \| R(u, \cdot) - R(v, \cdot) \|_2^2 \right),$$

(10)

where we have omitted the variable $\sigma$ as a matter of convenience and $\Omega$ is a normalisation constant given by

$$\Omega = \sum_{u,v \in G_k} K(L_k(u), L_k(v)) \exp \left( - \| R(u, \cdot) - R(v, \cdot) \|_2^2 \right).$$

(11)
In Equation (11), \( R(u,) \) is a vector spanning the whole set of wavelengths under consideration at a given scale and \( K(L_k(u), L_k(v)) \) is a kernel function.

Note that the formulation above is quite general and admits the use of a variety of kernel functions. Here, we explore the use of three different kernels. The first of these is the similarity measure discussed earlier based upon the cosine, i.e.

\[
K(L(u), L(v)) = \exp \left( - (1 - \alpha_{u,v}) \right)
\]

where \( \alpha_{u,v} \) is the cosine of the angle between the illuminants at adjacent pixels defined as

\[
\alpha_{u,v} = \frac{\langle L(u), L(v) \rangle}{\|L(u)\|\|L(v)\|}.
\]

The other kernel functions used here are the probabilistic variants for two popular M-estimators often used in robust statistics [41]. These robust estimators ensure the pixelwise illuminant statistics [41]. These robust estimators are not overly affected by unwanted outliers. These are the Huber estimator [41] given by

\[
K(L(u), L(v)) = \begin{cases} 
\exp \left( -\alpha_{u,v}^2 \right), & \text{if } |\alpha_{u,v}| \leq \tau \\
\exp \left( -2|\alpha_{u,v}| - \tau^2 \right), & \text{if } |\alpha_{u,v}| \geq \tau.
\end{cases}
\]

and the Tukey estimator [42]

\[
K(L(u), L(v)) = \begin{cases} 
\exp \left( -\frac{\tau^2}{6} \left( 1 - \left( 1 - \frac{\alpha_{u,v}}{\tau} \right)^2 \right)^3 \right), & \text{if } |\alpha_{u,v}| \leq \tau \\
\exp \left( -\frac{\tau^2}{6} \right), & \text{if } |\alpha_{u,v}| \geq \tau.
\end{cases}
\]

where, for both kernels, \( \tau \) is a cut-off variable.

Note that, in contrast with the non-probabilistic estimators used elsewhere, here we have used an exponential function to bound the kernels in Equations (14) and (15) above to the range [0,1]. We have done this without loss of generality following the notion that the likelihood in Equation (10) should satisfy the rules of probability.

D. Inference Process

With the probabilities above in hand, we proceed to solve the MAP inference problem in Equation (6). Note that the aim of computation in each subgraph is the set of marginal probabilities \( P(X_{g,k}(u)) \) for the pixelwise illuminant at the finest scale. In this paper, we employ the sum/max-product algorithm in [31], this is, effectively, a message passing strategy that entails exchanging information so as to update the probability distributions encountered throughout the graph.

Recall that the sum/max-product algorithm computes exact probability marginals of variables in a tree [43], [44]. In order to take advantage of this property, we construct our factor graph as a tree. Note that grid-graphs arising from lattices, such as the one used here, are often converted into trees using variable elimination techniques [45], [46]. These methods introduce fill-edges in lattice-like graphs, thus generating cliques with a large number of variables, i.e. with a large tree width. As a result, the computational burden can potentially increase greatly when the clique potential is computed. Moreover, the large tree width effectively draws many variables into a few cliques, changing the topology of the graph under study and potentially changing the inference problem in hand [47].

Consequently, we perform variable elimination using a convex hull algorithm for Delaunay triangulations [48]. We choose Delaunay triangulations motivated by the fact that they place adjacency constraints on the graph, i.e. the longest cycle in the graph is of length three. Thus, we have a limited treewidth\(^2\) which preserves the topology of the graph. This also ensures computational efficiency during message passing operations across the graph.

In Figure 4, we show the factor graph corresponding to the MAP problem in Equation (6). In the left-hand panel of the figure, we illustrate the interaction between variables across different scales after a Delaunay triangulation is effected. Note that this structure is consistent with the pyramid-like subgraphs in Figure 3. In the right-hand panel of Figure 4 we show the detail of the left-hand panel, where we have included the messages relayed between variables and factors. In the figure, we have adopted the notation used throughout the rest of the paper. The variables \( x \) and \( y \) denote the illuminant variables at different scales, respectively. The factor symbols are \( f \) and \( g \), and \( \mu \) indicates a message flowing in the direction of the sub-indexed arrow.

More explicitly, note the factor denoted by \( f \) “passes on” the probability information \( \mu_{f \rightarrow x} \) to the variable \( x \) according to the expression

\[
\mu_{f \rightarrow x} = \max_{\mathcal{X} \setminus x} F(\mathcal{X}) \prod_{y \in N_f \setminus x} \mu_{y \rightarrow f}.
\]

where \( \mathcal{X} = \mathcal{N}_f \) denotes the set of all variable nodes adjacent to the factor \( f \) and \( F(\mathcal{X}) \) is the probability distribution at the factor. Here, we have used \( \mathcal{X} \) purposely so as to make it clear that the quantities used to define the distribution \( F(\mathcal{X}) \) can be viewed as random variables. This is consistent with the notion that our message passing scheme entails computing probability distributions and marginalising across subgraphs. Consequently, the definition of the distribution \( F(\mathcal{X}) \) in Equation (16) at the factors is determined by the nature of the node, i.e. either inter-scale or within-scale. We illustrate these messages in the two left-most panels of Figure 5.

The messages from the variable node \( x \) to a factor node \( f \) are a product of neighbouring interactions between factors and the variable \( x \). Thus, the variable-to-factor message can be expressed as

\[
\mu_{x \rightarrow f} = \prod_{g \in \mathcal{N}_x \setminus f} \mu_{g \rightarrow x},
\]

where \( \mathcal{N}_x \) is the set of factor nodes neighbouring the variable node \( x \), and \( \mu_{g \rightarrow x} \) are the factor-to-variable messages defined in Equation (16).

In the right-most panel of Figure 5 we show the diagram for the variable-to-factor message passing in Equation (17). Note that, so far, we have focused or attention on the intra-scale

\(^2\) The treewidth of a graph is the number of variables in its largest clique.
and within-scale factor as defined in Equations (8) and (10). The inter-subgraph messages are derived from Equation (10), where neighbouring pixels are between two subgraphs rather than across scales. The explicit formulation is as follows,

\[
P(I_{k-1}(u), I_k(u) | L_{k-1}(u), L_k(u)) = \frac{1}{\Omega} K(L_{k-1}(u), L_k(u)) \times \exp \left( -\|R_{k-1}(u) - R_k(u)\|^2 \right),
\]

where, as before, we have omitted the variable \(\sigma\) and \(\Omega\) is a normalisation constant, \(K(L_{k-1}(u), L_k(u))\) is a kernel function and \(k\) is the index for the subgraph \(G_k\).

The distribution above can be substituted for \(F(\mathcal{X})\) in Equation (16) for purposes of message passing. Notice that the message of a given variable can also be computed using a sum-rule to marginalise posteriors in Equation (16) over the variables in \(\mathcal{X}\) and by excluding the variable \(x\) under consideration. After the messages have been sent in both directions on all edges in the graph, each factor is expected to contain the joint posterior probability distributions of the variables adjacent to the factor under consideration. As a result, in order to compute the marginal distribution for any one variable in the clique, we marginalise with respect to the adjacent variables making use of the joint distribution. As a result, the estimate \(L_{\sigma,k}(u)\) at a pixel \(u\) at the finest scale \(\sigma\) for a subgraph \(G_k\) can be viewed as the expected value of the illuminant with respect to the prototype set \(\Upsilon_k\).

Making use of Bayes’ rule and the notation in Section II-D, we can write

\[
P(\ell_i | L_{\sigma,k}(u)) = \frac{P(L_{\sigma,k}(u) | \ell_i) P(\ell_i)}{P(u)}
\]

(19)

The probability \(P(\ell_i | L_{\sigma,k}(u))\) in the equation above is, in practice, the posterior probability of a pixel being lit by the illuminant \(\ell_i \in \Upsilon_k\). This, in turn, is equivalent to the distribution \(P(L | G_k)\) in Equation (6).

Analogously, the most likely illuminant estimate \(L_{\sigma,k}(u)\) at a pixel \(u\) at scale \(\sigma\) for the subgraph \(G_k\) is computed using
Algorithm 1 - Pixelwise Illuminant Recovery

Require: \( I(u) \): image data; \( \sigma \in \Gamma \): set of Gaussian scales; \( b \): bandwidth parameter for the quick-shift method [36]; \( \tau \): kernel cut-off variable

Ensure: Illuminant estimate \( L(u) \)

\begin{algorithmic}[1]
  \State for all \( k \in K \) do
  \State for all \( \sigma \in \Gamma \) do
  \State Initialise illuminant estimate according to Algorithm 2 at scale \( \sigma \)
  \EndFor
  \State Construct graph \( G_k \) corresponding to the \( k^{th} \) colour constancy method under consideration
  \State Triangulate graph \( G_k \) using the convex hull technique in [48]
  \State for all Edges \( e \in E \) connecting variables in \( V \) and factors in \( F \) do
  \State Perform message passing using Equations (17) and (16)
  \EndFor
  \State Find \( L_{\sigma,k}(u) \) using the probabilities \( P(\ell_i|L_{\sigma,k}(u)) \) at the finest scale as shown in Equation (20).
  \EndFor
  \State return \( L(u) \) given by the geometric mean in Equation (21)
\end{algorithmic}

the illuminant’s weights in Equation (3). This is given by

\[ L_{\sigma,k}(u) = \frac{\sum_{\ell_i \in \Upsilon_k} P(\ell_i|L_{\sigma,k}(u))\ell_i}{\sum_{\ell_i \in \Upsilon_k} P(\ell_i|L_{\sigma,k}(u))}, \quad (20) \]

Here, we employ the finest scale so as to obtain the final illuminant estimate. As such, the final illuminant estimate is obtained by the geometric mean of the finest scale illuminants, \( i.e., \)

\[ L(u) = \left[ \prod_{G_k \in K} L_{\sigma^*,k}(u) \right]^{-|K|^{-1}}, \quad (21) \]

where \( \sigma^* \) is the finest scale for the \( k^{th} \) initialisation method under consideration.

III. IMPLEMENTATION

In this section, we elaborate further on the implementation of our method. In Algorithm 1, we show the step sequence of our method.

Note that our method, as described in Section II-B, requires a set of connected subgraphs each with strata representing multiple scales. For each of these subgraphs, a prototype set \( \Upsilon_k \) and initial illuminant variables \( L_{\sigma,k}(u) \) must be computed at initialisation. This is done in Line 3 of Algorithm 1, where we have introduced a call to Algorithm 2.

A. Initialisation

Algorithm 2 summarises how these prototypes and the initial illuminant estimates, denoted \( L^{(0)}_{\sigma,k} \) are computed using the Gaussian kernel in Equation (1).

Here, we commence by sub-dividing the image at each given scale into evenly sized patches. Next, we assume the illuminant to be uniform in each patch and use an existing colour constancy method to recover the illuminant for the patch under consideration. The choice of method is the same for all patches and all scales in each subgraph, but the methods differ from one subgraph to the other. Here, we have used 4 subgraphs so as to construct our graph, each of these corresponding to the estimates delivered by the grey world [9], white patch [12] and the 1\textsuperscript{st} and 2\textsuperscript{nd} order grey edge methods [11]. Once the initial estimates \( L_{\sigma,k}(u) \) for all scales \( \sigma \) of a subgraph \( G_k \) are in hand, we proceed to compute the prototypes \( \ell_i \) using the kernel mode seeking algorithm [36].

Thus, a complete graph will have four subgraphs, each of these comprised by strata corresponding to a scale \( \sigma \). Note that the structure of our graph is quite general and can employ other scaling schemes or colour constancy methods. Moreover, it is possible to implement more complex methods for the recovery of the prototype set for each graph as an alternative to the quick shift in [36]. Indeed, the prototypes may be recovered using the contour based region grouping method of Arbelaez et al. [49] or that in [50]. Moreover, pixelwise illuminant initial estimates can be obtained by considering small neighbourhoods about each pixel. Here, however, we have opted for a uniform grid-based approach so as to keep the computational burden low.

B. Illuminant recovery

Once the factor graph has been constructed and initialised, we proceed to compute a Delaunay triangulation across the three-level pyramid comprising each of the subgraphs \( G_k \). With the triangulated graph in hand, we then apply the message passing algorithm in Section II-D relying on the probability distributions in Equations (8) and (10). This is so as to solve the the MAP problem in Equation (6) and recover the marginals \( P(L_{\sigma,k}) \) at the finest scale of each subgraph. These marginals are then used as an additional “prior” in Equation (18).
Once the inference process has been performed on all subgraphs, the posterior probabilities $P(\ell_i | L_{\sigma,k}(u))$ from each subgraph can be used to compute the pixelwise illuminant using Equation (19) in Section II, where $\frac{P(\ell_i)}{P(\ell_{\eta})}$ is set to be a constant in Equation (20). This follows the intuition that every pixel and prototype are equally likely. With the posteriors in hand, the pixelwise illuminant can be computed in a straightforward manner using the geometric mean in Equation (21).

IV. Experiments

We now turn our attention to the utility of our method for pixelwise illuminant estimation of a single image and provide comparison to state of the art methods in literature. To this end, we have made use of four image datasets acquired under multiple light sources.

The first of these is the image dataset of natural outdoor scenes captured by Gijsenij et al. [20]. The ground truth colour for this dataset was acquired by positioning several grey surfaces on the scene and manually annotating the corresponding grey-world estimate for each illuminant. The second dataset used here is the multi-illuminant multi-object (MIMO) dataset presented in [28] using Equation (19). This comprises 78 images of scenes lit with multiple illuminants and taken in both controlled laboratory, and real-world outdoor environments. In contrast to the first dataset, the ground truth for this imagery was obtained per pixel, taking into account the contribution of the illuminants in the scene. The third dataset is the multi-illuminant image dataset of Bleier et al. [8]. It consists of four scenes acquired in two-illuminant lighting setups, where each lamp can bear several colour filters. The ground truth information for the dataset in [8] was acquired by painting the scene grey so as to suppress the surface albedo. Finally, we have also used a dataset of 135 images of five scenes acquired in house. Each scene is illuminated by two lights at a time, each of these configured in 27 different combinations of colour attained using Lee filters. Similarly to [8], we have obtained the illuminant colour ground truth by painting each scene with grey paint. In Figure 6 we show one sample image from each of the datasets used in our experiments.

For all our experiments, we have determined the parameter values for our method and the alternatives using cross-validation. For our dataset, we used a kernel bandwidth of 0.15 for the method in [36] to generate a small set of illuminant prototypes. For the datasets in [28] and that of Gijsenij et al. [20], we have set the bandwidths to 0.1 and 0.05, respectively. For the dataset of Bleier et al. [8], we have used a bandwidth of 0.2. Note that the robust kernels in Equations (14) and (15) require the cut-off variable $\tau$ to be set. For our dataset we have used $\tau = 0.75$, while for the the Beigpour et al. [28] and Gijsenij et al. [20] datasets, we employed $\tau = 0.8$ and $\tau = 0.9$ respectively. For the scenery of the Bleier et al. [8] we have set the value of $\tau$ to 0.9.

Here, we have compared our results against those yielded by a number of alternative methods. These alternatives are, the method of Gijsenij et al. [20], the multi-illuminant recovery method of Gu et al. [21] and the Multi-Illuminant Random Field (MIRF) algorithm presented in [28]. Note that these alternatives employ a variety of existing colour constancy methods as an integral part of the illuminant recovery process. Thus, the result yielded by these alternatives is dependent upon the colour constancy method used. This contrasts with our approach, which utilises all of them to initiate the subgraphs comprising our factor graph simultaneously and, hence, delivers a single result which accounts for the prototypes computed from all these methods. In this paper, to initialise our factor graph and perform experiments with the alternatives, we use the Grey-World (GW) [9], the first and second order Grey-Edge (1st GE and 2nd GE) [11] and the White Patch (WP) [12] algorithms.

For both our method and that of Gijsenij et al. [20], we have used a grid over the image lattice with a patch size of $10 \times 10$ pixels. Also, for the method in [20], we have set the number of the illuminants in the scene accordingly. In order to determine the optimal parameters of the MIRF algorithm of [28], we have applied tenfold cross-validation by randomly splitting the images of each dataset into two equally sized subsets for training and testing. Note that, while this method generates an arbitrary size of possible illuminants, it only estimates optimally two illuminants. This contrasts with our pixelwise estimation approach, which neither requires the number of illuminants to be known, nor restricts the illuminants to be the dominant pair of illuminants.

We commence by illustrating the effects of the initialisation choices on our method and showing sample results on the advantages of using multiple scales in our factor graphs. To this end, we have used two sample images in our dataset. In Figure 7, we show, in the top and third rows of the left-hand panels the input images, the ground-truth, i.e. the images where the object albedo has been removed by painting the scene grey, are shown in the rows below each of the input image. In the figure, we also show in the top-row of the right-hand columns, from left-to-right, the initial illuminant

\footnote{The dataset is accessible at http://colorconstancy.com/?p=659}
TABLE I

<table>
<thead>
<tr>
<th></th>
<th>Gu et al. [21]</th>
<th>Gijsenij et al. [20]</th>
<th>MIRF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st order GE</td>
<td>2nd order GE</td>
<td>GW</td>
</tr>
<tr>
<td>Our method</td>
<td>3.30 (2.68)</td>
<td>3.48 (2.96)</td>
<td>3.10 (2.59)</td>
</tr>
<tr>
<td>Gu et al. [21]</td>
<td>3.26 (3.25)</td>
<td>3.35 (3.16)</td>
<td>3.54 (3.22)</td>
</tr>
<tr>
<td>Gijsenij et al. [20]</td>
<td>7.17 (6.52)</td>
<td>6.81 (5.98)</td>
<td>6.26 (5.26)</td>
</tr>
<tr>
<td>MIRF</td>
<td>7.10 (4.70)</td>
<td>7.20 (5.00)</td>
<td>10.00 (10.10)</td>
</tr>
</tbody>
</table>

TABLE II
MEAN AND MEDIAN PER-PIXEL EUCLIDEAN ANGLE ERROR (IN DEGREES) FOR OUR METHOD AND THE ALTERNATIVES WHEN APPLIED TO THE BLEIER ET AL. [8] AND OUR DATASET. THE MEDIAN IS SHOWN IN PARENTHESIS AND THE ABSOLUTE BEST PERFORMANCE IS WRITTEN IN BOLD FONT.

<table>
<thead>
<tr>
<th></th>
<th>Bleier et al. [8] dataset</th>
<th>Our dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st order GE</td>
<td>2nd order GE</td>
</tr>
<tr>
<td>Our method</td>
<td>2.95 (2.90)</td>
<td>4.17 (3.34)</td>
</tr>
<tr>
<td>Gu et al. [21]</td>
<td>3.32 (3.39)</td>
<td>3.23 (3.32)</td>
</tr>
<tr>
<td>MIRF</td>
<td>7.60 (6.93)</td>
<td>7.37 (6.79)</td>
</tr>
</tbody>
</table>

Fig. 7. Left-hand column, first and third rows: Input image from our dataset; Left-hand column, second and fourth rows: Ground-truth images; Right-hand columns, on the top row, left-to-right: Initialisation illuminant maps based on the second-order grey edge [11], the grey world [9], and the white patch [12] methods; Corresponding bottom rows on the right-hand side, from left-to-right: The illuminant maps delivered by our method obtained using a single (finer) scale, those computed using multiple scales with only two initialisation methods (grey world [9] and the white patch [12]) and those obtained using, both the four initialisation algorithms and multiscale information.

estimates at the finest scale as delivered by the grey-edge [11], grey-world [9], and white-patch [12] methods. In the rows below the initialisation panels, we show, from left-to-right the output yielded by our method when a single scale is used for the initialisation provided by the three algorithms under consideration, when multiple scales are used on only the grey-world [9] and the white patch [12] methods, and, finally, when the four initialisation methods and multiple scales, are employed to recover the illumination map. From the figure, we can observe that, by adding both, multiple scales and initialisation schemes the illuminant maps clearly improve. Moreover, the results on the right-hand panels are less prone
to outliers introduced by the initialisations used in the figure. These observations are confirmed by the statistics reported in Table III. These show that the per-pixel mean and median Euclidean angle, in degrees, between the illuminant chroma delivered by our factor graph approach and the ground truth improves as multiple scales are introduced. In addition, when the Grey Edge [11] algorithms are used in combination with the Grey-World (GW) [9] and White Patch (WP) [12] methods for initialisation, the quality of the results also improves.

In Table I and II we present quantitative results for our method and the alternatives for the datasets under study. Note that our method achieves a single result with respect to the Grey-World (GW) [9], the first and second order Grey-Edge (1st GE and 2nd GE) [11] and the White Patch (WP) [12] algorithms. In the tables, we show the per-pixel mean and median Euclidean angle, in degrees, between the illuminant chroma delivered by our method and the alternatives with respect to the ground truth. Here, we opt for the Euclidean angle due to its widespread use in the colour constancy community [11], [20], [8], [28]. Furthermore, since the recovered illuminant and the ground truth are, by definition, normalised to unity, the Euclidean angle results shown here are correlated to the root-mean-square (RMS) error [21].

Note that our method out-performs all the alternatives when applied to both the Gijsenij et al. [20] dataset and our dataset. On the Bleier et al. [8] and MIMO datasets, our algorithm performance is comparable to the best results yielded by the alternatives. This is expected since our method uses all the colour constancy methods under consideration for purposes of initialisation, thereby delivering a unified result. Also, note that, in most cases, Gijsenij et al.’s method [20] does not perform as well as the others. This may be due to the assumption of a single constant illuminant at every sampled patch. Its white patch variant is, however, quite competitive over all the datasets. The MIRF method of [28] performs best on the MIMO dataset, with its results on other datasets often correlated with the method of Gijsenij et al. [20]. This is because the method in [28] obtains two dominant illuminants per scene, with each pixel being lit by a single light. The method in [21], despite being quite competitive, appears to show the largest variations with respect to the colour constancy method used.

Recall that our method employs the kernel $K(L(u), L(v))$ in equation (10) so as to compute the pixelwise illuminant. In Table IV, we show the mean Euclidean angle for the kernels described earlier in Section II-C. For purposes of further comparison, we have also included the results yielded by a uniform kernel given by

$$K(L(u), L(v)) = \frac{1}{\beta},$$

where $\beta$ is the number of colour constancy methods used to initialise our subgraphs, i.e. 4.

From Table IV, it is evident that the robust estimators provide a margin of improvement. This is expected, since pixels that deviate too far from the available illuminants are penalised accordingly. This is consistent with the bar plots in Figure 8, where we show the performance as a function of the datasets under study. From the figure, note that the Tukey estimator in Equation (15) consistently outperforms the Huber estimator in Equation (14). The only exception of this behaviour is the Bleier et al. [8] dataset. Further, the uniform kernel in Equation (22) delivers comparable results to those yielded by the cosine in Equation (12). This can be attributed to the prior term in Equation (10) favouring the contribution of the image data to the inference process.

It is worth noting in passing that, on average, our method takes approximately 125 seconds per image across the datasets under study. This is about 70% of the average per image timing for the MIRF algorithm [28], which took 170 seconds in average per image. In our experiments, the methods in [21] and [20] were the fastest to run, only taking an average of 30 seconds and 6 seconds per image, respectively. These timings, however, should be taken with caution since our method and those of Gijsenij et al. [20] and Gu et al. [21] have been implemented in Matlab. The MIRF algorithm [28] has been coded in C++. Moreover, the code for our method has not been fully optimised and improvements may be made accordingly. All our experiments were run on a workstation with 16 GB of RAM and an Intel Core i7-2600 CPU.

In Figure 9, we demonstrate the efficacy of our method and the alternatives for colour correction on real-world images corresponding to scenes that exhibit multiple illuminants. To this end, we present both, the illuminant maps yielded by all the methods under consideration and the colour corrected images computed using the recovered illuminant colour. In the figure, the left-hand panels show the input images whereas the right-hand columns show the results yielded by our method, the algorithm in [21] and the approaches in [20] and [28]. From the results shown in the figure we can appreciate that our method has successfully removed the “warm” effect of the lighting induced by the tungsten illuminating the front of either towers and the skylight in the background. Likewise, observe

<table>
<thead>
<tr>
<th>Single scale</th>
<th>Multiple scales</th>
</tr>
</thead>
<tbody>
<tr>
<td>GW, WP initialisation</td>
<td>Our method</td>
</tr>
<tr>
<td>Our dataset</td>
<td>5.96 (4.78)</td>
</tr>
</tbody>
</table>

**Table III**

**MEAN AND MEDIAN EUCLIDEAN ANGLE ERROR (IN DEGREES) OF THE ILLUMINANT MAPS IN FIGURE 7 WHEN USING A SINGLE (FINER) SCALE AND MULTIPLE SCALES WHEN ONLY TWO INITIALISATION METHODS ARE USED (GREY-WORLD (GW) [9] AND WHITE PATCH (WP) [12]) AS COMPARED TO OUR FACTOR GRAPH METHOD INITIALISED USING GREY-WORLD (GW) [9], THE FIRST AND SECOND ORDER GREY-EDGE [11] AND THE WHITE PATCH (WP) [12]. IN THE TABLE, THE MEDIAN IS SHOWN IN PARENTHESES.

<table>
<thead>
<tr>
<th>Co.s. dist. (Eq. (12))</th>
<th>Huber (Eq. (14))</th>
<th>Tukey (Eq. (15))</th>
<th>Uniform (Eq. (22))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gijsenij et al.</td>
<td>3.42/34</td>
<td>3.4922</td>
<td>3.9977</td>
</tr>
<tr>
<td>MIMO</td>
<td>3.4807</td>
<td>3.4807</td>
<td>3.8687</td>
</tr>
<tr>
<td>Bleier et al.</td>
<td>3.4284</td>
<td>3.4922</td>
<td>3.9977</td>
</tr>
</tbody>
</table>

**Table IV**

**MEAN EUCLIDEAN ANGLE ERROR (IN DEGREES) WHEN A NUMBER OF DIFFERENT KERNELS ARE USED IN OUR METHOD AS APPLIED TO THE DATASETS UNDER STUDY. ABSOLUTE BEST PERFORMANCE PER DATASET IS SHOWN IN BOLD FONT.**
that the sunlight reflections induced by the water on the boats in the second input image and the underwater scattering in the third input image have been both corrected using the illuminant map in the corresponding right-hand panels. Furthermore, in general, the illuminant maps delivered by our approach are in good accordance with the per-pixel changes expected from the input imagery. Although the results delivered by the methods in [21] and [20] exhibit heavy colour casts in places, they are also in good accordance with the per-pixel changes in the input imagery. In addition, the method of Gijsenij et al. [20] appear blocky. This is a result of the lattice-like strategy used by the method in [20]. Note that, for the second input image, the illuminant map delivered by our method suffers from a colour cast induced by the orange-brown sand whose chroma is almost identical to that of the illuminant. We also observe that, for the second scene, our illuminant estimation breaks down in the top-right and bottom-right corners of the image. This, is a result from the grey-edge initialisation, which, as expected, yields spurious results when no edge information is available in any given image patch.

V. CONCLUSION

In this paper, we have presented an approach to recover pixelwise illuminant colour for scenes with multiple illuminants which does not require precomputed libraries or user input. To do this, we have departed from the image formation process and formulated the illuminant recovery problem in a statistically data-driven setting based upon a factor graph. Our factor graph consists of a set of subgraphs, each of which employs the scale space of the input image and an existing colour constancy algorithm to perform statistical inference based upon a set of illuminant prototypes. This inference process is obtained by making use of a Delaunay triangulation on the graph. This treatment not only improves the inference, but also preserves the graph topology. We have performed a quantitative analysis of our method on four multiple illuminant datasets and compared against a number of alternatives. We have also evaluated the effect of varying the kernel function used by our method and demonstrated the effectiveness of our approach to perform colour correction of real-world images.

REFERENCES

Fig. 9. Left-hand column: Real-world input imagery; Right-hand columns from left-to-right: Results yielded by our method, the algorithm in [21] and the approaches in [20] and [28]. For each input image, the illuminant maps are shown in the top row, while the colour corrected images computed using the corresponing colour maps are in the bottom panels.


Lawrence Mutimbu  Lawrence Mutimbu received the B.Sc.(Hons) in Applied Mathematics from the National University of Science and Technology (NUST), Zimbabwe in 2008, both the M.Sc. in Industrial and Applied Mathematics from Eindhoven University of Technology (TU/e), Netherlands and the Diplom Ingenieur in Industrial Mathematics from the Johannes Kepler University (JKU), Austria in 2011 under the European Erasmus Mundus Programme. Currently, he is working towards his Ph.D. degree at the Australian National University and the National ICT Australia (NICTA). His research interests are graphical models for image processing, computer vision and pattern recognition.

Antonio Robles-Kelly  Antonio Robles-Kelly received the B.Eng. degree in electronics and telecommunications in 1998. In 2001 he received the William Gibbs/Plessey Award to the best research proposal and, in 2003, the Ph.D. degree in computer science from the University of York, UK. He remained at the University of York until December 2004 as a Research Associate under the Mathematics for IT (MathFIT) initiative of the Engineering and Physical Sciences Research Council (EPSRC). In 2005, he took a research scientist appointment with NICTA and, in 2006, he was appointed project leader of the Spectral Imaging project. In 2016 he took a Principal Researcher appointment at the Data61 business unit of CSIRO. He is also an adjunct Associate Professor at the ANU. He also serves as an Associate Editor of the IET Computer Vision and Pattern Recognition journals. His research interests are in the areas of computer vision, statistical pattern recognition, computer vision and image processing. Dr. Robles-Kelly has been a Chair, co-chair, and technical committee member of several mainstream computer-vision and pattern-recognition conferences.