A geometrical method of diffuse and specular image components separation

Ramón Moreno, Manuel Graña and Alicia d'Anjou

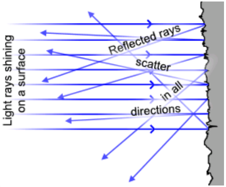
Computational Intelligence Group, University of the Basque Country
http://www.ehu.es/ccwintco

March 9, 2011

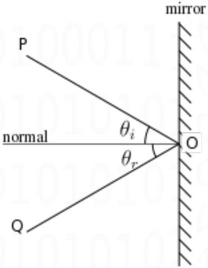
Introduction

- This approach is based on observed properties of the distribution of pixel colors in the RGB cube according to the Dichromatic Reflectance Model (DRM).
- We estimate the lines in the RGB cube corresponding to the diffuse and specular chromaticities.
- Then the specular component is easily removed by projection on the diffuse chromaticity line.
- The proposed algorithm does not need any additional information besides the image under study.

Diffuse and Specular reflections



(a)



(b)

(c)

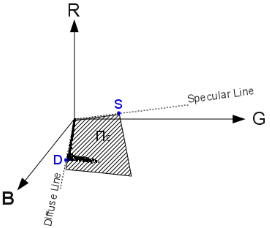
Figure: Diffuse reflection(a), specular reflection(b), natural image(c)

DRM

- The Dichromatic Reflectance Model (DRM) explains the formation of the image of the observed surface as the addition of a diffuse component D and an specular component S.
- Algebraically, the DRM is

$$I(x) = m_d(x)D + m_s(x)S$$

where m_d and m_s the diffuse and specular component weights respectively.



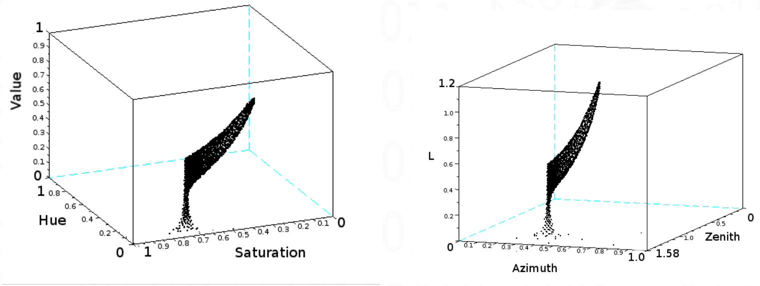
DRM

- In last figure the shaded region represents the convex region of the plane Π_c inside the RGB cube containing all the image colors resulting from the DRM equation.
- When there are several colors in the imaged scene, the DRM becomes $I(x) = m_d(x)D(x) + m_s(x)S$. Notice that D depends on the spatial coordinates x.

Color Space

- The DRM is defined as a vectorial sum in an euclidean space. This linearity exist in RGB however do not exist in other spaces like in the HSx family.
- When working with color images is very interesting to separate color in its components; intensity, chromaticity, hue and saturation specially when we are looking for photometric invariants.
- We can get it expressing the RGB color space by spherical coordinates, where a pixel $p_{euclidean} = \{r, g, b\}$ can be expressed equivalently in spherical coordinates by $p_{spheric} = \{\theta, \phi, l\}$ where θ, ϕ are the angular parameters and l is the vector magnitude.

Color Space



(a) (b)

Figure: Distribution of the ball image in the HSV color space (a) and in the spherical interpretation of the RGB color space (b)

Color Space

- Fig.2(b) shows the distribution of the image pixels in a $\theta - \phi - l$ space. As we can see the pixels distribution in this space is very close to the HSV space. But by difference with HSV, the spherical interpretation of the RGB color space let us express the DRM in spherical coordinates as

$$\mathbf{I}(\mathbf{x}) = (\theta_{\mathbf{D}}(\mathbf{x}), \phi_{\mathbf{D}}(\mathbf{x}), l_{\mathbf{D}}(\mathbf{x})) + (\theta_{\mathbf{S}}, \phi_{\mathbf{S}}, l_{\mathbf{S}}(\mathbf{x}))$$

  where the first one is the diffuse component and the second one the specular component.

- Then we can formulate this experiment working with the spherical interpretation, but always working in the RGB color space. In fact we trust in this approach for further works.

General description of the method

We assume that the observed surface is decomposable into patches of homogeneous chromatic characteristics. The proposed method has the following phases:

1. Chromatic line estimation: estimate the diffuse line L_d and the specular line L_s .
2. Dichromatization: We compute the parameters of the chromatic plane Π_{dc} in the RGB cube, and we project all the pixel colors into this plane. This step involves some additive noise removal.
3. Component separation: We compute the pure diffuse image component and the specular image component.

Chromatic line estimation

- We can easily appreciate the two main directions in the data. The most clear is the one corresponding to the diffuse line L_d which rises from the coordinate system origin.
- The second, less defined, appearing at the end of the diffuse elongation, is the specular direction identified by the specular line L_s.

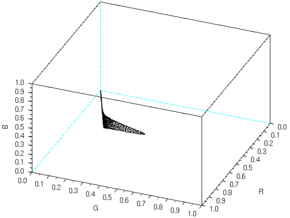


Figure: Synthetic image plotted in the three-dimensional RGB space

Chromatic line estimation

- We perform a Principal Component Analysis (PCA) which give us the direction of the chromatic line $\overrightarrow{u}$.
- Therefore the diffuse chromatic line is defined as

$$L_d : (r, g, b) = P + s\overrightarrow{u}; \forall s \in \mathbb{R}$$

- Analogously, we select the brightest pixels, obtaining a mean point Q in the RGB cube and the largest eigenvector $\overrightarrow{v}$ for the specular color, therefore the specular chromaticity line is expressed as follows

$$L_s : (r, g, b) = Q + t\overrightarrow{v}; \forall t \in \mathbb{R}$$

Image dichromatic regularization

- Once we know the chromatic lines, we build the dichromatic plane Π_{dc} in $\mathbb{R}^3$ which is the best planar approximation to the color distribution in RGB.
- It can be expressed as follows: $\Pi_{dc} : (r, g, b) = P + s\overrightarrow{u} + t\overrightarrow{v}; \forall s, t \in \mathbb{R}$, and the normal vector is $\overrightarrow{N} : \overrightarrow{u} \times \overrightarrow{v}$, where $\times$ denotes the conventional vector product.

Image dichromatic regularization

- To remove noise and regularize the image colors we project the pixel's colors into this dichromatic plane Π_{dc} .
- For each image point color in the RGB cube I_i we compute the line $L_i : (r,g,b) = I_i + k\vec{N}; \forall k \in \mathbb{R}$, which is orthogonal to the dichromatic plane Π_{dc}, and to regularize I_i we compute its projection I_i^c as the intersection of L_i with Π_{dc}.

Component separation

Recalling the DRM definition $I(x) = m_d(x)D + m_s(x)S$ our goal is to bring the pixels to the chromatic line, that is $\forall x : m_s(x) = 0$.

- We proceed as follows: for each regularized image point I_i^c lying in the plane Π_{dc} we draw the line

  $$L_i : (r, g, b) = I_i^c + t\overrightarrow{v}; \forall t \in \mathbb{R}$$

  where $\overrightarrow{v}$ is the specular line vector director.
- The pixel diffuse component corresponds to the intersection point I_i^d of this line with the diffuse line

  $$L_d : (r, g, b) = P + s\overrightarrow{u}; \forall s \in \mathbb{R}$$

  and it exists because they lie in the same plane Π_{dc} and they are not parallel lines.

Component separation

- We have obtained $I^d(x) = m_d(x)D$ so that $\forall x,: m(x) = 0$, and the resulting image $I^d(x)$ is purely diffuse, without specular components.
- Obtaining the specular image component is then trivial if we recall the DRM definition:

$$I^s(x) = I(x) - I^d(x) = I(x) - m_d(x)D = m_s(x)S$$

.

Experimental results

- The first is a synthetic image (using Blender), and the second is a natural image.
- We know the original surface color ($r = 0.790$, $g = 0.347$ and $b = 0.221$) in the synthetic image, we can compute an estimation of the error committed by our estimation of the diffuse image. If we denote Q the original color, the error is the distance of this point to the diffuse line, computed as $d(Q, L_d) = ||\vec{PQ} - \perp(\vec{PQ}, \vec{u})||$, where $\perp(\vec{a}, \vec{b})$ denotes the projection operator.
- In the images shown in figure 4 the error committed is 0.0116. Variations in the error are due to the diffuse region pixel selection.

Experimental results

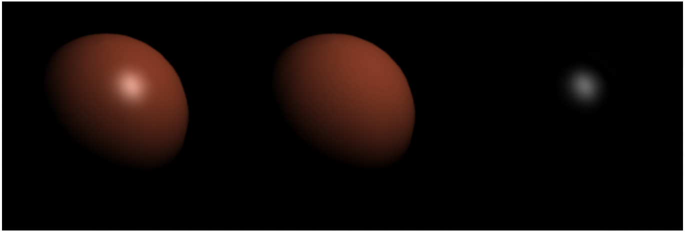


Figure: Synthetic image, diffuse image and specular image

Experimental results

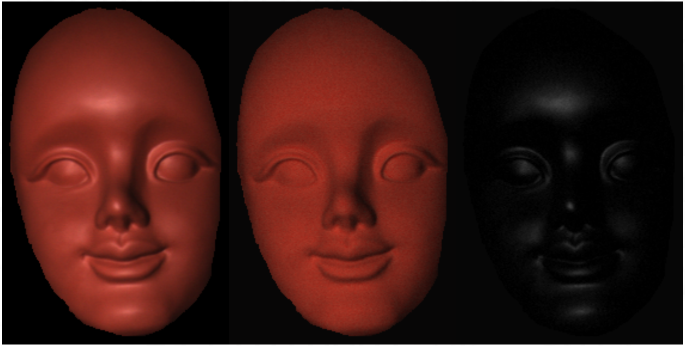


Figure: Natural image, diffuse image and specular image

Conclusions

- We have described an image component separation for monocolor images which is very effective, fast and robust.
- It has been developed from the DRM and is well theoretically grounded despite its simplicity.
- It consists in the estimation of the diffuse and specular lines as the principal components of diffuse and specular point clouds, respectively, selected from the image by hand.

Conclusions

- Contrary to other approaches our approach does not need specific hardware devices, and only needs one image.
- Our approach does not need a Specular Free image, it provides almost simultaneously both image components.
- On going work is addressing the extension of this approach to images containing several surface colors, i.e. $I(x) = m_d(x)D + m_s(x)S$, and to images with illumination sources of different colors, i.e. $I(x) = m_d(x)D + m_s(x)S(x)$.

Thanks so much for your attention.

Time for questions