Vision, Modeling, and Visualization (2010)

Gerbil–A Novel Software Framework for Visualization and Analysis in the Multispectral Domain

J. Jordan & E. Angelopoulou

Pattern Recognition Lab, University of Erlangen-Nuremberg, Germany

Abstract

Multispectral imaging has been gaining popularity and has been gradually applied to many fields besides remote sensing. Multispectral data provides unique information about material classification and reflectance analysis in general. However, due to the high dimensionality of the data, both human observers as well as computers, have difficulty interpreting this wealth of information. We present a new software package that facilitates the visualization of the relationship between spectral and topological information in a novel fashion. It puts emphasis on the spectral gradient, which is shown to provide enhanced information for many reflectance analysis tasks. It also includes a rich toolbox for evaluation of image segmentation and other algorithms in the multispectral domain. We combine the parallel coordinates visualization technique with hashing for a highly interactive visual connection between spectral distribution, spectral gradient and topology. The framework is released as open-source, has a modern cross-platform design and is well integrated into existing established computer vision software (OpenCV).

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Photometry

1. Introduction

In Multispectral imaging, the camera captures the scene with a higher spectral resolution in comparison to traditional color cameras. Each image pixel holds a vector of intensity values (instead of an R, G, B triplet), where each value corresponds to the incoming light over a small wavelength range. This may be achieved by capturing each band seperately with narrow color filters or by scanning the image with a spectrometer. Until recently, multispectral imaging has been mainly employed in remote sensing. It has also been gaining popularity in the fields of agriculture and cultural heritage. However, technological advances in multispectral sensor design are creating sensors that are faster and less expensive than before. This makes it feasible to use multispectral data for traditional scene and reflectance analysis. Furthermore, this analysis has an impact on tri-chromatic image capture, since it helps in developing new theories on how to best interpret RGB data.

Several software packages exist today that explicitly deal with the processing and interpretation of multispectral data [ITT10, BL02, Uni10, JM03, LFH08, RH08]. However, most of them have a very application-specific interface and functionality. A large percentage of them is expressly designed for remote sensing. In addition to several preprocessing steps, e. g. undistortion or data range normalization, the typical focus of such software lies on classification, which is often based on semantic information. The classification results are presented using false coloring, or in specific plots of data classes. A significant portion of the information contained in the image (other than material) is often either ignored or treated as noise. Thus, it is not comprehensively presented to the user.

To properly analyze the reflectance of a scene, it is vital to understand how light, geometry and material interact with each other. The different influences that lead to the observed reflectance are hard to separate by inspecting single pixels, or only statistical measures over a larger region of pixels. Instead, spectral variations within a region and/or within the same material class, often reflect the sought information. Current visualization is based on the view of single bands and statistical plots (e. g. presenting the mean and variance values for a region). However, a complete visualization of the spectra of all pixels in the region is more informative. For example, by

DOI: 10.2312/PE/VMV/VMV10/259-266

examining neighboring pixels simultaneously, one may more easily observe the effects of light source position relative to a scene object, or be able to separate texture effects. Figure 1 illustrates how the effect of several reflectance phenomena can be easily distinguished in such an examination format.

We propose a new paradigm for visual assistance of multispectral analysis that specifically addresses the lack of a seamless integration of spectral distribution and topology. Our software, called *gerbil*, allows the simultaneous visualization of different types of spectral plots, as well as the output of image analysis algorithms. Gerbils (the animals) are known as hardy, efficient (in terms of locating and using food) and people friendly creatures. Our gerbil platform exhibits similar attributes. It is powerful, efficient in both time complexity and information discovery, and user-friendly. Besides the informative visualization framework, gerbil:

- provides tools specifically designed for reflectance analysis (spectral gradient)
- incorporates state-of-the-art segmentation techniques
- is well integrated with existing computer vision software
- has a modern cross-platform design
- will be released as open-source.

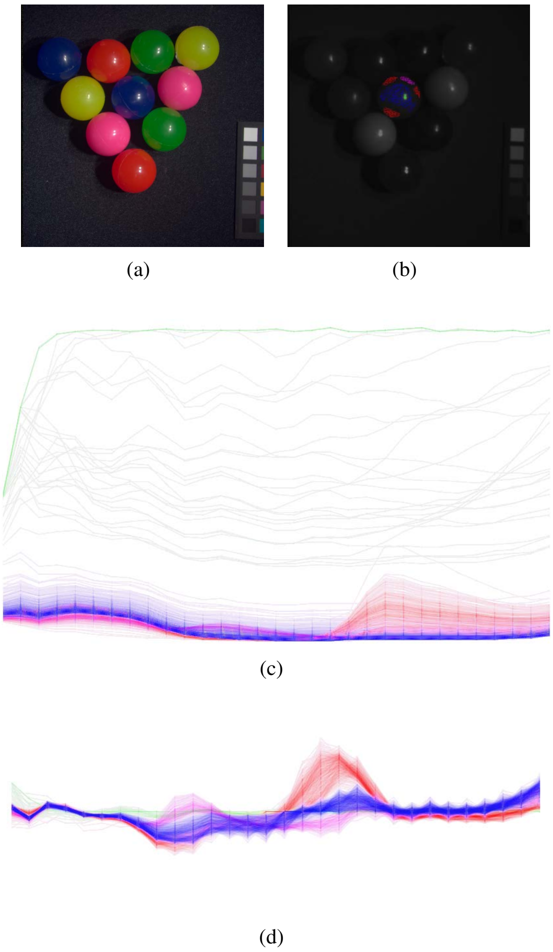
2. Previous Work

Software that processes multispectral data has a long history. However, most often it is found to be very specific to astronomical or remote sensing applications. A 2007 review of publicly available software for processing hyperspectral remote sensing image data is given by Larry Biehl [Bie07]. Below, we give a short overview on several software solutions that have relevance today.

A popular commercial tool that handles multispectral images is the ENVI software [ITT10] from ITT Visual Information Solutions. It includes powerful modules especially for remote sensing applications. However, its support of visual inspection is restricted to viewing single bands of the image or false-coloring.

MultiSpec is a freeware package by Larry Biehl and David Landgrebe [BL02]. It can analyze multispectral images from various sources and puts particular emphasis on classification. It is being developed since the early 1990s and was still maintained as of early 2010. The software is mainly targeting general Earth science community; it does not provide a view of the spectral distributions. The user can generate plots of mean and standard deviation in the spectral domain, or a 1D histogram from a selection of pixels. In 2008, Roberto and Hofer presented Theia, a software also directed at multispectral image analysis [RH08]. The major contribution according to the authors is a more modern software design and efficient handling of the data. However, it does not provide any improved functionality over software like Multispec.

HyperCube, as released by the U.S. Army Geospatial Center, contains functions to filter, warp (register two images), calibrate and undistort, photogrammetrically project, and arithmetically manipulate multi- or hyper-spectral data [Uni10]. However, its various visualization options do not directly tackle the multispectral nature of the data, apart from allowing one to scroll through image bands one by one.


(a) (b)

(c)

(d)

Figure 1: *Reflectance phenomena and their effect on the spectral distribution of a single object (blue ball).* ***(a)*** *RGB view of the image.* ***(b)*** *Manually labeled pixels in one spectral band.* ***(c)*** *Spectral distributions of labeled pixels.* ***(d)*** *Spectral gradient distributions of labeled pixels. The different colored labels correspond to:* ***Blue:*** *Primarily diffuse reflectance.* ***Green:*** *Primarily specular reflectance.* ***Red:*** *Inter-reflections from red materials.* ***Magenta:*** *Inter-reflection from green materials. Images were obtained by gerbil.*

For astronomical data, a very powerful imaging and data-visualization software is available from the Smithsonian Astrophysical Observatory, called ds9 [JM03]. It is the successor to SAOtng, which dates back to 1995. One of its main focal points is scale algorithms and color maps for the visualization of one or multiple frame buffers. It also allows binning and filtering of data. Another one of its key features is its ability to draw contour overlays. Unfortunately, the narrow scope of ds9's feature set makes it of limited use to other applications. Also focused on astronomical data is the more recent approach of Li et al. [LFH08]. Bands are drawn in 3D space either as an image-stack or as a volume-rendered model. Clutter is reduced by applying transparency to individual values based on their intensities. While their visualization can also be useful for analysis of multispectral data in general, it is a different approach than ours which focuses on topology.

A software framework for multispectral analysis that originated in the reflectance analysis community is the MultiSSA and MultiSFAMS packages released by Angelopoulou et al. in 2005 [APK10]. Amongst other features, they provide spectral gradient computation and unsupervised mean-shift segmentation on multispectral data. However, the software does not provide any means of interactive visualization.

While there exists a range of high quality, well-maintained software which visualizes multispectral images and/or provides a comprehensive toolset for data analysis, the current state-of-the-art lacks: a) much-needed universality, and b) support of reflectance analysis on multispectral data.

3. Spectral Gradient

Multispectral images provide a wealth of information that allows one to better understand the complex behavior of image formation. Most existing software focuses on segmentation and classification. One of the goals of gerbil is to provide more powerful reflectance analysis tools. An important step in that direction is the introduction of the spectral gradient as an additional feature for visualization.

The intensity values we obtain from a photosensitive sensor correspond to the incident irradiance on the area of a pixel on the sensor plane. In the notation of [Ang00, Ang07], the amount of light reflected from each point $\boldsymbol{p} = (x,y,z)$ in the scene results from a combination of the light incident on $\boldsymbol{p}$ (determined by its spectrum $e(\lambda)$ and its direction $E(\boldsymbol{p})$), and the surface reflectance $S(\boldsymbol{p},\lambda)$ of the surface at $\boldsymbol{p}$:

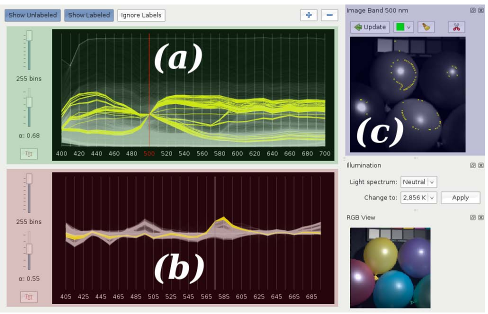
$$I(\boldsymbol{p},\lambda) = e(\lambda) \cdot E(\boldsymbol{p}) \cdot S(\boldsymbol{p},\lambda) \quad , \tag{1}$$

where λ denotes wavelength. The reflectance function $S(\boldsymbol{p},\lambda)$ depends on the surface material, the scene geometry and the viewing and incident angles. The spectral derivative that forms the spectral gradient is defined as the partial derivative of the logarithmic image with respect to the wavelength λ,

$$L_\lambda(\boldsymbol{p},\lambda) = \frac{\partial(ln(I(\boldsymbol{p},\lambda)))}{\partial\lambda} \quad . \tag{2}$$

According to [Ang00], for diffuse surfaces, the spectral derivative is the normalized partial derivative of the surface albedo $\rho(\boldsymbol{p},\lambda)$ offset by a constant illumination term c:

$$L_\lambda(\boldsymbol{p},\lambda) \approx \frac{\rho_\lambda(\boldsymbol{p},\lambda)}{\rho(\boldsymbol{p},\lambda)} + \frac{c}{e(\lambda)} \quad , \tag{3}$$



Light spectrum: Neutral

Change to: 2,856 K

Figure 2: *Gerbil user interface.* ***(a)*** *Spectral distribution view.* ***(b)*** *Spectral gradient distribution view.* ***(c)*** *Spatial view of a single image band.*

where $\rho_\lambda(\boldsymbol{p},\lambda) = \partial\rho(\boldsymbol{p},\lambda)/\partial\lambda$ is the partial derivative of the albedo with respect to wavelength. Albedo itself is a material property, independent of illumination and geometry.

Similarly, for purely specular reflectance, the spectral derivative becomes the normalized partial derivative of the Fresnel term offset by the same constant illumination term [Ang07]:

$$L_\lambda(\boldsymbol{p},\lambda) \approx \frac{F_\lambda(\boldsymbol{p},\lambda)}{F(\boldsymbol{p},\lambda)} + \frac{c}{e(\lambda)} \quad . \tag{4}$$

where $F_\lambda(\boldsymbol{p},\lambda) = \partial F(\boldsymbol{p},\lambda)/\partial\lambda$ is the partial derivative of the Fresnel term with respect to wavelength. The Fresnel term is also a material property, but unlike albedo it also depends on geometry [FvDFH95].

The spectral gradient is the discrete approximation of spectral derivatives obtained by finite differencing. By definition, spectral gradients are a powerful tool for image analysis because:

- In purely diffuse surfaces, the spectral gradient is invariant to geometry. When the illuminant spectrum does not change over the topology of the image, this means that spectral gradient values stay constant within a material.
- Specular regions in the image can be separated from diffuse ones using the spectral gradient.

Often the spectral gradient is better suited for analysis of the captured scene when compared to the original spectral data. In the visual inspection of an image, the spectral gradient feature-vectors can offer a view that focuses more on data aspects that are directly related to material properties and the image formation process (for more details please refer to [Ang07]). Therefore, spectral gradients are also incorporated into the visualization.

4. Gerbil

The goal of gerbil is to help the user with: a) the interactive inspection of multispectral images and b) the evaluation of algorithms applied on them. We want to present the wealth of information that a multispectral image contains in a meaningful, comprehensible way. The graphical user interface is depicted in Figure 2. It concentrates around three key elements, which we refer to as viewports:

1. The spectral distribution view of the current image. It shows the spectral vector distributions of (a) the entire image, or (b) several disjoint pixel sets in the image, which are distinctly colored for improved discernability.
2. The distribution view of the corresponding spectral gradient as defined in Eq. 2, with the same functionality.
3. One or several dockable/floating windows that depict user-selected bands of the image or its spectral gradient. They may also show an RGB representation of the image.

A powerful aspect of the software is how these are combined. The user has several tools at hand to work on the different viewports simultaneously. Tracking of the mouse cursor results in real-time updates in all viewports. A labeling of data can be performed in each viewport, and propagated to the others. Such a workflow helps the user to explore the data step-by-step and is recognized as a valuable tool for understanding complex data [FH09]. This is further explained in Section 4.2 below, after Section 4.1 describes the technical details behind the spectral distribution views.

As previously mentioned, the user may work with pixel labels. Labels are vital for differentiation and comparison, and do not necessarily need to be created by the user manually. Instead, state-of-the-art segmentation algorithms can be employed. Segmentation not only assists the user in labeling data, but also plays a key role in several types of scene analysis. We integrate both supervised and unsupervised segmentation methods. These work either with or without topological information (i. e. based solely on the density of the image data). The integration of these algorithms is discussed in Section 4.3.

Multispectral imaging is a helpful tool in studying effects of different lighting on various computer vision algorithms. It can also reveal the effects of metamerism in human perception and RGB data. Gerbil assists in analysis related to these topics through the application of different illuminants to the image data (see Section 4.4). An RGB image is then created based on human visual perception (see Section 4.5).

The framework is written according to modern coding standards and software design paradigms. The C++ code builds on the powerful and mature Qt toolkit from Nokia [Nok10]. This provides a good foundation for platform-agnostic development. We maintain the software on all major operating systems, including GNU/Linux-based systems as well as Mac OS X and Windows. Furthermore, processing of the image band data is done through the well-established OpenCV computer vision software library [Bra00], which enables the easy integration of common image analysis algorithms. Section 4.6 explains how the internal data structure helps in extending the framework.

4.1. Spectral Distribution View

In the spectral distribution view, the information available in the entire image (or a subset of its pixels) is presented at once. To display the high-dimensional spectral vectors, we employ the Parallel Coordinates [ID90] method as explained below. It is a well established technique for visualizing high-dimensional geometry and analyzing multivariate data, e. g. in financial applications or geographic information systems. It is based on connecting multimodal data points. The traditional spectral visualization is just a specific instantiation of parallel coordinates visualization. By employing this method we can now incorporate the state-of-the-art tools for high-dimensional data presentation.

The D-dimensional feature space (resulting from D spectral bands) is projected onto a two dimensional view as follows. D parallel vertical lines denote the D axes, i.e. the D spectral bands. To display the spectral vector of a pixel, a polyline is drawn with its vertices lying on the vertical axes. The y-coordinate of the vertex on the ith axis corresponds to the value of the pixel at band i. When this is done iteratively for all pixels, the display may easily get cluttered and single polylines may not separate well from the rest of the data. However, coloring and transparency, as well as interactively highlighting specific data of interest, significantly improve the perceptibility of the data. Additionally one can use clutter reduction techniques for parallel coordinates (e. g. [ED06]).

Next to clutter, drawing speed becomes an important issue when dealing with multispectral images of high spatial resolution. We significantly reduce the amount of polylines drawn by histogramming. A D-dimensional histogram (each dimension reflecting a spectral band) is built with N evenly distributed bins in each dimension. N is user-adjustable between 2 and the dynamic range of the captured data (typically 2^8 or 2^{16}). The histogram is visualized using the aforementioned parallel coordinates. For each non-zero entry a polyline with varying opacity α is drawn. α is determined by the relationship between the number of pixels lying in the bin, n_{bin}, and the total number of processed pixels, n_{total}. To avoid the case that sparsely filled bins are hard to perceive, the logarithm is applied:

$$\alpha = \frac{\log(n_{bin}+1)}{\log n_{total}} \quad . \tag{5}$$

Building such a sufficiently fine-grained histogram in the multispectral data space is not feasible in general. However, we can rely on the spatial resolution as a strong limiting factor to the histogram denseness. A concrete example is the publicly available CAVE multispectral image database [YMIN08]. It provides images with 31 bands of 16 bit intensity data. For such data the histogram covers a space of $2^{16 \cdot 31}$ possible values. If we partition each band into 256 bins, we end up with a total of 2^{248} bins. However, the histogram is very sparse. The reason is, that the spatial resolution of the image restricts the total amount of different data points we may observe. An image from the CAVE database has a spatial resolution of 512×512, leaving only 2^{18} possible values. Therefore, the histogram can be built and read in a time and space efficient manner through the use of an ordinary hashing algorithm. Hash collisions only rarely occur.

The technical requirements for this operation are moderate. The software can be operated interactively on a current desktop machine, or even a mobile computer. To achieve this, OpenGL is employed for drawing all viewports. The user is given control of the number of bins N per dimension, and can therefore find a good trade-off between detail of view and drawing speed. On a laptop with Intel Core 2 Duo 2.93 GHz CPU and Nvidia Quadro FX 2700M mobile GPU, the CAVE datasets can be conveniently analyzed. When downsampled versions of the same images with a resolution of 128x128 pixels are used, no noticeable lag occurs, even when setting N to the maximum.

4.2. Interactive Inspection

We tackle the lack of a seamless integration of spectral distribution and topological relationship in two steps. First, we implement an efficient visualization of the spectral distribution, as was explained above. Second, we provide rich human interaction that intuitively combines spatial (band) and spectral view. The user can interactively transfer information between one view and the other in both directions.

An initial step of intuitively exploring data is to grasp individual portions of it. Instant feedback to a mouse-driven manual selection of specific image pixels (or data points in the spectrum) gives valuable assistance in understanding the formation of the captured data. To make this possible, a suitable highlight, or overlay, in yellow color is drawn in the spectral and spatial viewports. Highlights and overlays are updated simultaneously and in real-time, while the mouse movement is tracked. In a spectral viewport, the mouse can be used to highlight all pixels that fall into a particular value-range in a specific band. It can also be used in dynamically introducing and modifying range limits in each band. The former can instantly reveal the effect of different spectral clusters on an observed intensity. The latter is a less convenient, yet more powerful method of picking specific spectra. In both methods, the highlight in the spectral viewport is realized by drawing the polylines associated with the selection of histogram bins in yellow and fully opaque. In the spatial viewports it is displayed by drawing a yellow cross over all pixels that fall into the selected bins. When the mouse is navigated over the spatial viewports, the spectrum of the pixel under the cursor is displayed as an overlay in the corresponding spectral viewports.

For a more thorough analysis, highlights need to be made permanent. For this, the user can choose from a set of five different labels, which are identified by all primary and secondary colors except yellow. This number was chosen as more labels at once would be rarely needed and the choice of colors makes them most discernible. Label colors are applied on the grayscale data in the spatial viewports, as well as the drawing operations in the spectral viewports. To draw different labels in the spectral viewports, a separate sparse histogram is computed for each label. As discussed above, the user may interactively highlight specific spectra with the mouse, which then appear in yellow. Accordingly, if the user highlights labeled spectra, a touch of yellow is added to their label color to make highlights distinguishable.

The current overlay, whether it originated from a spectral or a spatial viewport, can be added to or subtracted from any label set. Each image pixel can belong to one label set at a time. This enables the user to iteratively refine labels according to either topology or spectrum. The user can then, not only examine the whole spectral distribution of a specific region at once, but also compare different portions of the scene in the spectral view. This is a vital tool in manually distinguishing different materials or reflectance phenomena (see Figure 1). An additional benefit provided by gerbil is the employment of segmentation algorithms for labeling.

4.3. Segmentation

In a typical inspection session, the user often needs to mark several regions of the image, e. g. objects, or specularities, to compare them against others. A valuable tool would then be the possibility to automatically separate image regions according to initial user input. Recent work in the area of seeded image segmentation includes graph cuts, random walker, and shortest path optimization algorithms. In 2009, Couprie et al. introduced the Power Watersheds framework [CGNT09]. In that framework, a general formulation is proposed that combines several seeded segmentation methods.

Our software provides access to these methods as follows: From user-provided seeds, a bipartitioning (foreground, background) of the image is performed. The user seeds image regions by marking foreground and background exemplar pixels. The algorithm performs the bipartitioning either on (a) the currently visible band, or (b) the spectral vectors of the image, or (c) the spectral gradient vectors. The resulting foreground pixels are then associated with the current label, which may already hold other pixels, therefore enabling the consecutive composition of a label set.

In order to segment the multispectral image, existing algorithms need to be extended to work on multispectral input. The segmentation problem is posed as a graph problem. A graph is constructed where each node represents an image pixel. Nodes are connected by edges based on the image topology (typically a 4-connected lattice). The weights assigned to an edge denote the similarity between its endpoints. Couprie et al. [CGNT09] propose to set the weight w_{ij} for an edge between nodes i and j to

$$w_{ij} = \exp(-\beta(I_i - I_j)^2) \quad , \tag{6}$$

where I_i is the 1D pixel intensity (e. g. of a grayscale image) corresponding to node i and β is a free parameter. This can be adjusted to multispectral pixel vectors by using either the Euclidean or the χ^2 distance. In the latter case,

$$w_{ij} = \exp\left(-\beta(\sum_{k=1}^{D} \frac{(I_i^k - I_j^k)^2}{I_i^k + I_j^k})\right) \quad . \tag{7}$$

The Euclidean distance is a natural first choice that is widely used in RGB and multispectral images. χ^2 was introduced in gerbil in the hopes of reducing the effects of noise. The intensity differences obtained in each band are analyzed in relation to the intensity values themselves.

Performing the segmentation in the multispectral domain has both advantages and disadvantages. Multispectral segmentation is a harder task compared to the 1D case as the distance measurement suffers from the high dimensionality of the data [BGRS99]. On the other hand, the segmentation may more accurately detect borders in the image that are not found by analyzing isolated spectral bands. In experimentation we found that segmentation accuracy often profits from the additional information in the whole image data, or spectral gradient data compared to single band intensities.

Another interesting aspect of multispectral data analysis is then clustering or segmentation of the image in the spectral domain. A well-established unsupervised clustering method that does not take topology into account is the mean-shift density gradient estimator [CM02]. In gerbil, we include Multi-SFAMS [APK10], a software package that already adapts mean-shift to multispectral data.

4.4. Scene Lighting

An important aspect in reflectance analysis is the illumination of the scene and how it can affect appearance. According to Eq. 1 (see page 3) the captured spectrum $I(\boldsymbol{p},\lambda)$ is formed by a combination of the illumination spectrum $e(\lambda)$ and geometrical and material effects, which are independent of $e(\lambda)$, $R(\boldsymbol{p},\lambda) = E(\boldsymbol{p})S(\boldsymbol{p},\lambda)$. Different illuminants result in changes of the captured intensities. This can pose problems to computer vision algorithms based on color cues, e. g. tracking, stereo, image retrieval. In multispectral data, assuming homogeneous scene illumination, the illuminant can be exchanged by a straightforward calculation. The effects of such a change can be presented to the user within our spectral visualization framework.

If the spectrum $e(\lambda)$ of the incident light is known, the normalized reflectance of a pixel becomes: $R(\boldsymbol{p},\lambda) = I(\boldsymbol{p},\lambda)/e(\lambda)$. Available multispectral images are often already normalized with respect to illumination and sensor sensitivities. In either case, a new illuminant spectrum $e'(\lambda)$ can then be applied by setting $I'(\boldsymbol{p},\lambda) = R(\boldsymbol{p},\lambda) \cdot e'(\lambda)$.

Due to the spectral sampling, the intensity of a certain band does not contain the information of a single wavelength but the integral over a range of wavelengths. Therefore, applying these calculations to the image bands is only an approximation. However, these approximations can be justified by the typically narrow filter bandwidth of multispectral sensors and the widely-accepted observation that the spectra of most surfaces and illuminations are smooth functions.

Several reference illuminants are available for image relighting. We model them as black body radiators [WS00]. They, in turn, can be described by Planck's law, where the illuminant color is parametrized by the color temperature T in Kelvin. Daylight has a color temperature $T \approx 6500\,K$, while tungsten light bulbs have a temperature $T \approx 2800\,K$.

4.5. Human color perception

Research on multispectral data is not disconnected from the research on traditional image formation, or human visual perception. Multispectral data capture can help to obtain insights that also apply to traditional RGB color images. To assist in that research and also make the examination more intuitive to the user, we implement methods to produce RGB output according to the human perception of color. The resulting RGB image is incorporated into the interactive user interface.

The transformation from a spectrum to its RGB values is done via the CIE XYZ color space [WS00]. The transform to CIE XYZ requires the CIE standard observer functions, which are defined according to human color perception. They model the light sensitivity of the three types of cones in the human eye. Transforming CIE XYZ values to RGB colors involves the definition of a reference white. Gerbil uses the sRGB color space with reference white D65, as it is specifically designed for computer displays [SACM96].

4.6. Extensibility

For some applications it may be necessary to introduce new functionality to the framework. Hence, the source code of the software is readily available under a free software license. Equally important, gerbil provides a powerful, easy to use programming interface to the image data. It is built on an internal representation of multispectral images that combines two different data structures as described below. These enable convenient, yet efficient access for reading and writing. Furthermore, they are applicable to different acquisition methods, e. g. tunable filters, push broom sensors, etc.

1. As is common for storage (and most often capture), a multispectral image is represented by an array of intensity images, each describing a specific spectral band. This representation has the advantage that any algorithm suitable for an intensity image can be applied on the band data individually. The widely adopted OpenCV code-library provides a rich set of functions that are commonly used in computer vision [Bra00]. Therefore, we let OpenCV handle the image bands and use the OpenCV API to create and access them.

2. For many applications including ours, it is necessary to work with the full spectral information on a per-pixel basis. It is, therefore, preferable to have the whole pixel data available in memory in a continuous fashion. The existing memory layout would become suboptimal, as it would prevent hardware pre-fetching. This can eventually lead to highly degraded computational performance [Dre07]. Hence, we also store an array of pixel vectors representing the image (redundant pixel cache). This array is accompanied by a "dirty bit" map, where each bit defines whether the corresponding vector is still valid or not. Depending on the application, the pixel cache may be updated on write, or as soon as a vector is accessed.

This internal format of a multispectral image makes it easy to extend the software with new functionality. Specific bands or pixels can easily be read and written back using a simple C++ API. Many image operations can be directly performed by the OpenCV library, without necessitating data duplication or even conversion. Note that OpenCV itself provides images with more than three, and by default up to 64 channels. This limit is hard-coded. The number of channels must be known at compile time. It is also often cumbersome in OpenCV to work on selected channels only.

Our software design allows the user to incorporate his own algorithms into the software, leveraging both the OpenCV functionality, and the input/output, data handling and visualization functionality of the gerbil framework.

5. Usage Examples

For a better insight on the user interface of Gerbil, we present two example uses of gerbil. For this, we use images from the CAVE multispectral image database [YMIN08].

Materials: Figure 3 shows the result of segmenting various fake and real foods. In the spectral view, the effects of geometry on the band intensities is clearly visible. However, the differences between fake and real pepper could be mistakenly attributed to brightness only. The spectral gradient helps to easily distinguish between the different materials by omitting geometrical effects.

Reflectance: In Figure 4, diffuse and specular regions are compared. A closer examination of the segmented specularities (red and blue) shows that the spectral gradient reveals the material-dependency of the Fresnel effect. In order to indicate this effect all pixels that exhibit a specific intensity at 650 nm are highlighted. Note the smooth transition in the gradient plots from purely specular pixels (blue) to those exhibiting a mixture of diffuse and specular reflectance (green).

6. Conclusions

We presented a new visualization framework for multispectral data that is highly interactive and allows for thorough analysis. Given its intuitive interface, it is also well suited for educational purposes. Gerbil enables a new workflow in visual inspection of multispectral images due to: (a) its real-time view of spectral distributions, spectral-gradient distributions, and single band images and/or an RGB representation of the image; (b) the strong interactivity of these viewports and their connection, where the user can highlight and label data in a feedback cycle; and (c) the integration of supervised and unsupervised segmentation algorithms.

The software as presented is released under a free software license at `http://sf.net/p/gerbil`. Thus, researchers are able to apply it on their own analysis tasks and extend it with further functionality as seems fit. This adaptability is fostered by the internal design of the software, its embedding of the major existing computer vision library and platform independence.

References

[Ang00] ANGELOPOULOU E.: Objective colour from multispectral imaging. In *European Conference on Computer Vision* (2000), Springer-Verlag, pp. 359–374.

[Ang07] ANGELOPOULOU E.: Specular highlight detection based on the fresnel reflection coefficient. In *IEEE International Conference on Computer Vision* (2007).

[APK10] ANGELOPOULOU E., PARANJAPE M., KOTHAVADE T.: MultiSSA: MultiSpectral Scene Analysis Software. http://www5.cs.fau.de/fileadmin/Persons/AngelopoulouElli/MultiSSA, Jun 2010.

[BGRS99] BEYER K., GOLDSTEIN J., RAMAKRISHNAN R., SHAFT U.: When is "nearest neighbor" meaningful? *Database Theory–ICDT '99* (1999), 217–235.

[Bie07] BIEHL L.: Eval-Ware: Hyperspectral Imaging [Best of the web]. *IEEE Signal Processing Magazine 24*, 4 (2007), 125–126.

[BL02] BIEHL L., LANDGREBE D.: MultiSpec–a tool for multispectral-hyperspectral image data analysis. *Computers & Geosciences 28*, 10 (2002), 1153–1159.

[Bra00] BRADSKI G.: The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[CGNT09] COUPRIE C., GRADY L., NAJMAN L., TALBOT H.: Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In *IEEE International Conference on Computer Vision* (2009), pp. 731 –738.

[CM02] COMANICIU D., MEER P.: Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (2002), 603–619.

[Dre07] DREPPER U.: What every programmer should know about memory. *http://people.redhat.com/drepper/cpumemory.pdf* (Nov 2007).

[ED06] ELLIS G., DIX A.: Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics* (2006), 717–724.

[FH09] FUCHS R., HAUSER H.: Visualization of multi-variate scientific data. In *Computer Graphics Forum* (2009), vol. 28, pp. 1670–1690.

[FvDFH95] FOLEY J., VAN DAM A., FEINER S., HUGHES J.: *Computer graphics (in C): principles and practice*. Addison-Wesley Longman, 1995.

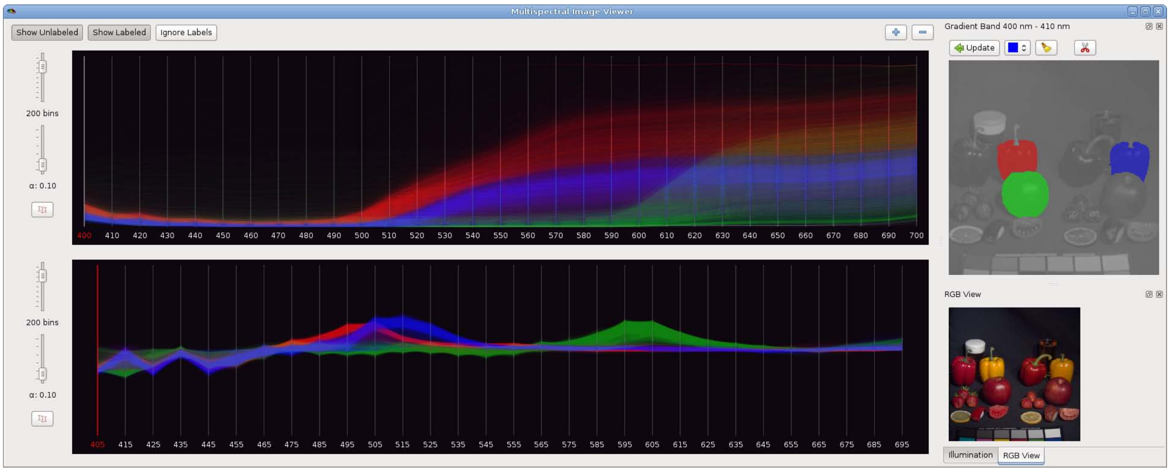


Figure 3: *Screenshot of gerbil working on the* Fake and Real Food *image. Labels obtained by graph cut segmentation.* ***Blue:*** *Real pepper.* ***Red:*** *Fake pepper.* ***Green:*** *Fake apple.*

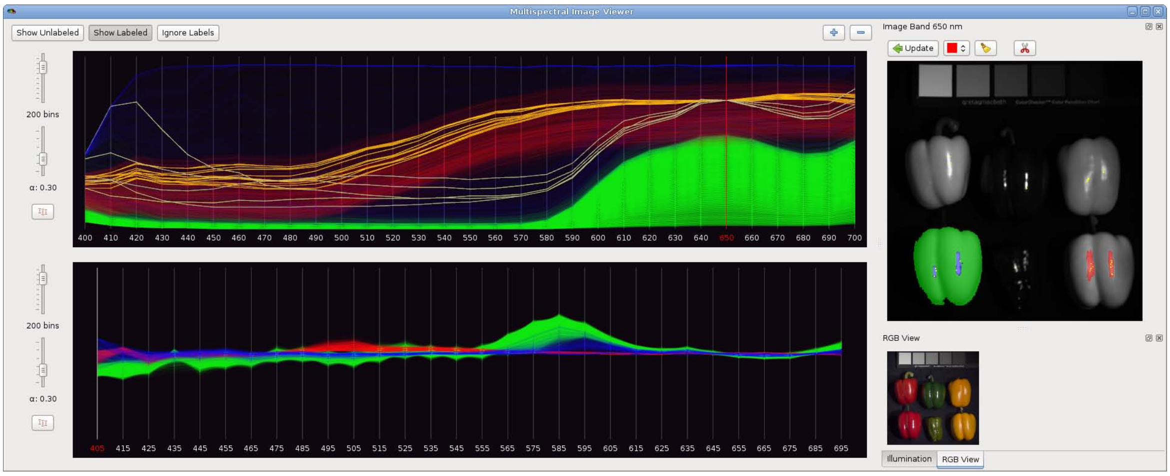


Figure 4: *Screenshot of gerbil working on the* Fake and Real Peppers *image. Labels obtained by graph cut segmentation.* ***Green:*** *Primarily diffuse reflectance on red pepper.* ***Blue:*** *Primarily specular reflectance on red pepper.* ***Red:*** *Primarily specular reflectance on yellow pepper. The highlight shows how different specular regions are separable by their spectrum.*

[ID90] INSELBERG A., DIMSDALE B.: Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *IEEE Conference on Visualization* (1990), pp. 361–378.

[ITT10] ITT VISUAL INFORMATION SOLUTIONS: Envi. http://www.ittvis.com/ProductServices/ENVI.aspx, Jun 2010.

[JM03] JOYE W. A., MANDEL E.: New features of SAOImage DS9. In *Astronomical Data Analysis Software and Systems XII* (2003), vol. 295, pp. 489–492.

[LFH08] LI H., FU C.-W., HANSON A.: Visualizing multiwavelength astrophysical data. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1555–1562.

[Nok10] NOKIA: Qt - cross-platform application and UI framework. http://qt.nokia.com/, Jun 2010.

[RH08] ROBERTO V., HOFER M.: Theia: open environment for multispectral image analysis. In *International Working Conference on Advanced Visual Interfaces* (2008), ACM, pp. 462–465.

[SACM96] STOKES M., ANDERSON M., CHANDRASEKAR S., MOTTA R.: *A Standard Default Color Space for the Internet-sRGB*. Tech. Rep. http://www.color.org/sRGB.xalter, Microsoft and Hewlett-Packard, Nov 1996.

[Uni10] UNITED STATES ARMY GEOSPATIAL CENTER: Hypercube. http://www.agc.army.mil/Hypercube/, Jun 2010.

[WS00] WYSZECKI G., STILES W. S.: *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2 ed. Wiley-Interscience, July 2000.

[YMIN08] YASUMA F., MITSUNAGA T., ISO D., NAYAR S.: *Generalized Assorted Pixel Camera: Post-Capture Control of Resolution, Dynamic Range and Spectrum.* Tech. rep., Columbia University, Nov 2008.