

A framework for storage, visualization and analysis of multispectral data

Stephan Irgenfried¹ and Christian Negara²

¹ Karlsruhe Institute of Technology KIT,
Institute of Process Control and Robotics IPR,
Engler-Bunte-Ring 8, D-76131 Karlsruhe

² Fraunhofer Institute of Optronics, System Technologies and Image
Exploitation, Fraunhoferstr. 1, D-76131 Karlsruhe

Abstract In this paper we describe our database centered workflow for acquisition, enrichment, long-term storage and web-based analysis of multi- and hyperspectral image data and measurement metadata. We propose a standardized way of storing large amounts of measurement data using a hybrid approach consisting of a relational database and direct file access through a common data access layer. Data import and export is performed by either using proprietary file formats like ENVI or by using an universal XML-based data format. To allow preview and analysis of image data, a web-based application has been developed which supersedes the need for client-side installation of tools like MATLAB to perform spectral or spatial analysis of the data while still allowing third-party applications to retrieve data from the database to perform in-depth analysis like automated filter design. Successful implementation of the workflow is demonstrated by the example of rapid application development for mineral sorting.

1 Introduction

While classification of materials based on their spectral reflectance properties has been an area of research in remote sensing applications for a long time, it is attracting more and more attention in the area of industrial applications, e.g. sorting materials as part of the recycling process, quality assurance in food production or enrichment of primary

resources. Common to those applications is the need to measure the reflectance spectrum of the materials to be detected using appropriate equipment and to analyze the acquired data to define classification features and optical filters. To be able to build productive and robust applications, the whole process from sample selection and spectral measurement as well as data storage and analysis up to filter and application design has to be taken into account and is preferably standardized. In this work we propose a workflow for this process, describe our implementation and present results. We especially focus on a solution for the long term raw measurement data storage, which tackles the problem of the loss of measurement information over time as described by Michener and Brunt [1]. The rest of the article is organized as follows: in Chapter 2 related work is discussed. Chapter 3 describes our approach and the implementation details. In chapter 4 we finish with a summary and an outlook on future work.

2 Related work

Acquiring, storing and analyzing hyperspectral information has a long history in remote sensing applications. A couple of databases have been set up to make hyperspectral material information publically available, e.g the ASTER spectral library [2], USGS digital spectral library [3], hyperspectral.info [4], SPECCHIO [5], Vegetation Spectral Library [6]. Also background work on the design and implementation of hyperspectral databases, metadata and measurement process was published by some authors, e.g. Ruby [7] or Pfitzner [8, 9], who introduced 12 rules to be taken into account when measuring and storing hyperspectral data. There are some very good analysis and visualization tools for multispectral data available like the Environment for Visualizing Images (ENVI) [10] and Gerbil [11]. Also the use of MATLAB [12] is very common. While all these programs are desktop applications and mostly used by experts in the image processing or statistical domain, our software is designed to provide some basic visualization tools, e.g the mean spectral reflectance and its standard deviation, to anybody via a user-friendly web application replacing the need for installation of dedicated analysis software on the user computer.

3 Our work

Within our work we focus on design and implementation of a consistent workflow for hyperspectral data to improve long term availability of measurement data and to support analysis and application development. Fig. 19.1 gives a basic overview of our framework which will be described in more detail in the following chapters.

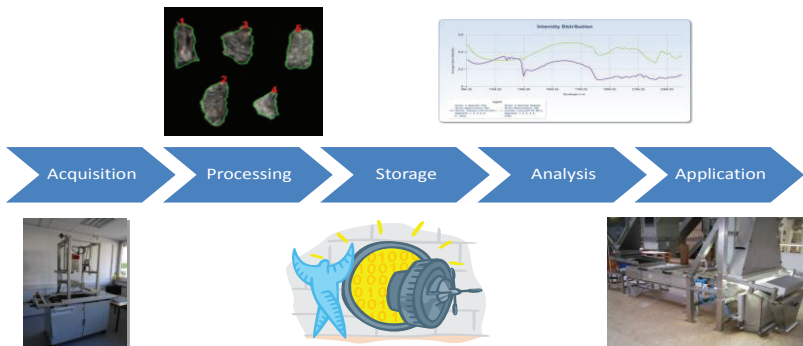


Figure 19.1: A consistent workflow from data acquisition to application development.

3.1 Data acquisition

The data acquisition is performed in the lab at the Fraunhofer IOSB with different types of sensors: There are high speed and high resolution RGB cameras, single-channel UV/NIR cameras and low speed and low resolution hyperspectral cameras ranging from 240-400nm, 366-720nm and 1000-2500nm. The data processing chain consists of the following semi-automatic steps to prepare measured data to be imported to the database:

White balance, outlier detection and normalization

To get comparable results, the spectra have to be normalized. This is obvious when using different measurement stations, but also has to be

done for measurements using only one spectrometer, because the illumination might change. The spectra are normalized to the interval $[0, 1]$ because the sensors have different dynamic ranges. The white balance is computed in two steps. For every hyperspectral image I , an image I_w of a shading bar is also acquired. I is a function that maps x, y, λ to an intensity value, whereas x, y define the pixel position and λ the wavelength. Because in our lab we only use line sensors for multispectral data acquisition, the x -value defines the position on the line, the y -value defines a time-line which results from moving the probe on a linear table through the viewing plane of the camera. After subtracting the dark current, the following formula is used to compute the white balanced image I' from the original image I :

$$I'(x, y, \lambda) = \frac{I(x, y, \lambda)}{\bar{I}_w(x, \lambda)} \cdot \frac{t_w}{t} \quad (19.1)$$

$$\bar{I}_w(x, \lambda) := \frac{\sum_y I_w(x, y, \lambda)}{\sum_y 1}$$

t, t_w are the respective exposure times. The normalization function $\bar{I}_w(x, \lambda)$, which is computed from the image of the shading bar, is invariant of y because the illumination effects don't change during the movement of the linear table. In the literature some further normalizations like the spectral gradient [11] or the division by the cumulative intensity [13] are applied to remove the effects of the illumination or object geometry. Some of these normalizations are not always applicable, because they assume a hyperspectral image as input data, but we often only use up to four discrete wavelength bands in our sorting machines. Another drawback is the loss of the cumulative intensity information, which can be an important feature for classification, especially when only a few bands are measured.

The image I' is normalized according to the shading bar. Therefore the spectrum has to be divided by the reflectance spectrum $R(x, \lambda)$ of the shading bar:

$$I''(x, y, \lambda) = \frac{I'(x, y, \lambda)}{R(x, \lambda)} \quad (19.2)$$

Outlier detection is another important preprocessing step. The hyperspectral sensors may have some defect pixels which are known and return invalid (e.g. constant) intensity values. But also over- and under-exposed pixels with intensities near the extrema of the dynamic range have to be marked as invalid. To be able to use formula (19.1), the intensity has to be linearly dependent from exposure time. Hence, depending on the sensor characteristics, upper and lower thresholds exist, which define the valid intensity range. All other pixel values in I are marked as invalid and ignored in later steps.

Registration

If the probe is measured with different sensors (e.g. at different wave length ranges), the multispectral images have to be registered to be able to compute correlations between these ranges. This is done via a semi-automatic algorithm written in MATLAB. First, the user marks some corresponding object points in the multispectral images. Because the spectrum ranges of the sensors do not necessarily overlap, the intensity values can't be directly compared. Therefore, the two images are first converted to edge-images using a sobel filter. If E_x is the edge-image resulting from using a vertical sobel filter and E_y from using a horizontal sobel filter, the orientation independent gradient magnitude image is computed by:

$$E = \sqrt{E_x^2 + E_y^2} \quad (19.3)$$

For every band of the multispectral images a corresponding edge-image is computed and afterwards the edge-images are averaged. The two resulting grayscale images are normalized to the interval [0,1] because different materials can induce small differences in one spectrum but big difference in the other. The transformation matrix between the two hyperspectral images is computed with MATLAB by maximizing the cross correlation value of the grayscale edge-images. Because we deal only with line scan sensors the search space of the transformation matrices can be reduced to affine transformations.

Segmentation

The resulting image of a measurement typically contains objects of the same material or different materials as shown in Fig. 19.4. The objects are segmented using a multimodal segmentation algorithm based on the algorithm described by Boykov and Funca-Lea [14]. All bands of the registered images are compound into a single stacked image. The algorithm works as follows:

- The user marks pixels which definitely belong to the background and pixels which belong to the foreground
- An energy minimization algorithm computes a foreground/background separation of the pixels in the image
- To remove very small foreground objects, a pixel is only marked as foreground if the pixels in the neighborhood are also foreground pixels. The neighborhood is defined by a mask with variable size.
- The foreground/background separation is used to number the objects in the image by computing the connected components of the foreground pixels.
- The user has the possibility to manually change the object numbering or assign objects to the background.

The foreground/background separation problem can be described as the MAP (maximum a posteriori) solution x^* of a markov random field:

$$p(x|y = \hat{y}) = p(x|\hat{y}) = \frac{p(\hat{y}|x)p(x)}{p(\hat{y})} \quad (19.4)$$

$$x^* = \max_x p(x|\hat{y})p(x) \quad (19.5)$$

$x = (x_0, \dots, x_n)$ is a random vector which contains for every pixel a labeling $x_i \in \{Foreground, Background\}$. $\hat{y} = (\hat{y}_0, \dots, \hat{y}_n)$ is the observed data. y_i is the spectrum of pixel i . $p(\hat{y})$ is a constant and can be discarded when computing the MAP. We use a 4-connected graph for the markov random field. Hence (19.4) can be factorized as:

$$p(x|\hat{y}) = \frac{1}{Z} \prod_i \psi(\hat{y}_i, x_i) \prod_{i,j \in \mathcal{N}} \phi(x_i, x_j) \cdot \frac{1}{p(\hat{y})} \quad (19.6)$$

The set \mathcal{N} contains all edges of the graph, Z a normalization constant, and ϕ, ψ are interaction potentials. Applying log on both sides we obtain:

$$x^* = \max_x (\log p(x|\hat{y})) = \min_x (-\log p(x|\hat{y})) \quad (19.7)$$

$$= \sum_i \underbrace{-\log \psi(\hat{y}_i, x_i)}_{=:E_d(x_i, \hat{y}_i)} + \sum_{i,j \in \mathcal{N}} \underbrace{-\log \phi(x_i, x_j)}_{=:E_s(x_i, x_j)} \quad (19.8)$$

The data energy potential $E_d(x_i, \hat{y}_i)$ characterizes the compliance of the labeling of a pixel to its observed spectrum. If an arbitrary pixel in the image has a foreground label associated, then it should have a similar spectrum to the points, which were manually marked as foreground. The similarity of a pixel's spectrum to the spectra of the marked pixels is computed via k -NN search in \mathbb{R}^s where s is the number of bands. The spectrum of every pixel x_i corresponds to a point $S(x_i)$ in \mathbb{R}^s which is obtained from the intensity values of every band. For every pixel in the image the $k = 50$ marked pixels are searched, whose spectra match best i.e. have the smallest euclidean distance. $E_d(x_i = l, \hat{y}_i)$ is then set to the number of pixels whose label is l divided by k while $l \in \{\text{Foreground}, \text{Background}\}$. Instead of using the original spectra $S(x_i)$, the normalized spectra $\tilde{S}(x_i) = S(x_i)/\|S(x_i)\|$ are used. The cumulative intensity is discarded. This is needed because a human individual often marks pixels, which can be clearly recognized as foreground or background. Pixels at the border of the object are usually not marked, because their intensities are low due to poor illumination, unless an integrating sphere is used. By dividing by the cumulative intensity, all pixels of the same material have spectra which are very near in \mathbb{R}^s independent of the surface orientation and therefore the cumulative intensity at those pixels.

The smooth potential $E_s(x_i, x_j)$ has to be a metric. The smooth potential has the following form:

$$E_s(x_i = l_i, x_j = l_j) = \begin{cases} 0 & , \text{ if } l_i = l_j \\ \frac{\alpha \|S(x_i) - S(x_j)\|^2}{2\sigma^2} & , \text{ otherwise.} \end{cases} \quad (19.9)$$

The term $\|S(x_i) - S(x_j)\|^2/2\sigma^2$ is derived from the logarithm of a normal distribution and σ is the sensor noise which is set to $1/256$. This

value is derived from the quantization error of a 8 bit sensor. σ could also be assessed from the sensor specification. In contrast to the computation of E_d , the original spectra are used here. α is set to $1/5000$ to clearly prefer a low data potential over a low smooth potential because it proved to be more robust if hyperspectral images with many bands are used. x^* is computed using the graph-cut algorithm with α -expansion described in [15–17], which can be applied because E_s is a metric.

3.2 Data storage

Our main goal for the database is to develop a long term storage and sharing platform for spectral signatures of materials. A key feature is to store the full measurement raw data to work with arbitrary analysis software directly on it.

To avoid storing a large amount of raw pixel data in the relational database, causing it to grow fast and loose performance [18], we decided to use a hybrid approach to store multispectral image data and metadata, shown in Fig. 19.2. While keeping measurement data in com-

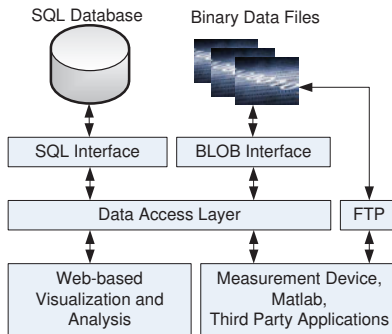


Figure 19.2: System overview of data storage and access.

compact binary files as created by the measuring device and accessing it with a fast binary data access library we store data that benefits from SQL Server query optimization and constraint checking in a relational database.

Compared to storing the measurement data and metadata in a filesystem structure, our approach of storing measurement data and metadata has several advantages:

- Increased flexibility using a data access layer. The data access layer allows changes to the database or the binary files without requiring changes on the client side.
- Improved access rights handling. Access rights to the data can be handled inside the data access layer. File system level permission handling is no longer required. In addition, database access can be regulated in fine detail.
- Metadata is stored in a relational database. Measurement metadata is stored in a relational database for improved query performance and data integrity checking. The metadata consists of 4 main groups of data:
 - Measurement related data e.g. materials, material groups, system operator, measurement device, project
 - Image related data e.g. object segmentation, file type, scaling information, color space.
 - Hardware related data e.g. spectrometer characteristics, white balance information, illumination parameters
 - Project related data e.g. projects, sites, operators, access rights

3.3 Data visualization and analysis

Another key feature of our application is the web-based visualization and analysis of multispectral measurement data. Besides a standard web browser no client side software installation is required to be able to browse the multispectral image data and the metadata in the database. Standard analysis functionality includes e.g. intensity distributions or histograms of different materials, different material conditions or different measurements of a single material. Measurement series can be compared and the results are visualized as 2D-charts (Fig. 19.3). The web-based visualization is also capable of displaying the raw image data acquired by the camera as shown in Fig. 19.4. An intuitive slider control panel allows to quickly access a particular wavelength image

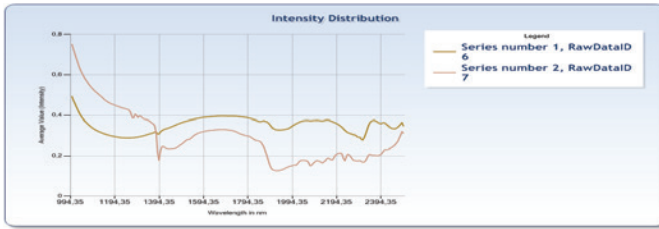


Figure 19.3: Intensity distribution for two different materials in the NIR and SWIR spectrum.

and data. The user can choose between a preview of the original image or showing only the labeled objects, hiding background and other objects. An additional feature shown in Fig. 19.4 is the possibility to create an RGB false color coded image by assigning intensity values of particular wavelengths to the R-,G- and B-channel of the resulting image thus allowing to better visualize spectral behaviour of the material. Based on our new mass storage infrastructure, the web-based visualiza-

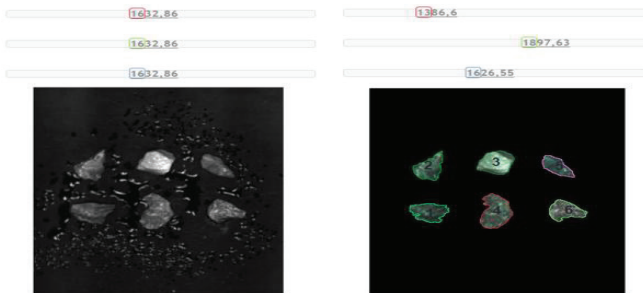


Figure 19.4: (Left) Preview of an intensity image for a selected wavelength including background and surrounding objects. (Right) False-Color display of labeled objects only with an overlay of object borders and object numbers.

tion and analysis allows fluent browsing through the database content and fast image and data preview while keeping load on database and webserver hardware as low as possible. The application is implemented using ASP.Net MVC.

4 Summary and outlook

We proposed a workflow and an infrastructure to acquire, enrich, store, analyze and retrieve multi- and hyperspectral measurement data. A working prototype implementation was presented. We're currently investigating how to improve binary data storage performance by using an object database.

Acknowledgements

Some of the software has been implemented as part of the student project ViMuDat. The authors thank the students Eva Wolkwitz, Angie Neumann, Niko Fellner, Marcus Gall, Anton Chalakov and Vivek Sharma for their valuable contributions.

References

1. W. K. Michener, J. W. Brunt, J. J. Helly, T. B. Kirchner, and S. G. Stafford, "Nongeospatial metadata for the ecological sciences," *ECOLOGICAL APPLICATIONS*, vol. 7, no. 1, pp. 330–342, 1997.
2. A. Baldrige, S. Hook, C. Grove, and G. Rivera, "The aster spectral library version 2.0," *Remote Sensing of Environment*, vol. 113, no. 4, pp. 711–715, 2009.
3. R. N. Clark, G. A. Swayze, R. Wise, E. Livo, T. Hoefen, R. Kokaly, and S. J. Sutley, "Usgs digital spectral library splib06a: U.s. geological survey, digital data series 231," Denver and CO and USA, 2007.
4. J. G. Ferwerda, "A free online reference library for hyperspectral reflectance signatures: A web-based interface for reference spectra of natural surfaces facilitates data sharing and increases access to reference material for classification," *SPIE Newsroom*, 2006.
5. A. Hüni, J. Nieke, J. Schopfer, M. Kneubühler, and K. I. Itten, "The spectral database specchio for improved long-term usability and data sharing," *Computers & Geosciences*, vol. 35, no. 3, pp. 557–565, 2009.
6. Vegetation Spectral Library, "Vegetation spectral library." [Online]. Available: <http://spectrallibrary.utep.edu/>
7. J. G. Ruby and R. L. Fischer, "Spectral signatures database for remote sensing applications," in *Imaging Spectrometry VIII*, ser. Proceedings of SPIE, S. S. Shen, Ed. Seattle and WA and USA: SPIE, 2002, vol. 4816, pp. 156–163.

8. K. Pfitzner, R. Bartolo, Byan B., and A. Bollhöfer, "Issues to consider when designing a spectral library database," in *Spatial intelligence, innovation and praxis*, Spatial Sciences Institute, Ed., Melbourne, 2005.
9. K. Pfitzner, A. Bollhöfer, and G. Carr, "A standard design for collecting vegetation reference spectra: Implementation and implications for data sharing: Journal of spatial science," *Journal of Spatial Science*, vol. 51, no. 2, pp. 79–92, 2006.
10. Exelis Visual Information Solutions, "Envi." [Online]. Available: <http://www.exelisvis.com/ProductsServices/ENVI.aspx>
11. J. Jordan and E. Angelopoulou, "Gerbil - a novel software framework for visualization and analysis in the multispectral domain," in *Proceedings of the Vision, Modeling, and Visualization Workshop 2010*, R. Koch, A. Kolb, and C. Rezek-Salama, Eds. Eurographics Association, 2010, pp. 259–266.
12. E. Arzuaga-Cruz, L. Jimenez-Rodriguez, M. Velez-Reyes, D. Kaeli, E. Rodriguez-Diaz, H. Velazquez-Santana, A. Castrodad-Carrau, L. Santos-Campis, and C. Santiago, "A matlab toolbox for hyperspectral image analysis," in *Proceedings of the Geoscience and Remote Sensing Symposium, 2004. IGARSS '04.*, vol. 7, 2004, pp. 4839–4842.
13. W. Kessler, *Multivariate Datenanalyse : für die Pharma-, Bio- und Prozessanalytik; ein Lehrbuch*. Weinheim: Wiley-VCH, 2007.
14. Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," *Int. J. Comput. Vision*, vol. 70, no. 2, pp. 109–131, Nov. 2006. [Online]. Available: <http://dx.doi.org/10.1007/s11263-006-7934-5>
15. Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1222–1239, 2001.
16. V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 147–159, 2004.
17. Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 9, pp. 1124–1137, sept. 2004.
18. R. Sears, C. van Ingen, and J. Gray, "To blob or not to blob: Large object storage in a database or a filesystem?" 17.06.2006. [Online]. Available: <http://research.microsoft.com/pubs/64525/tr-2006-45.pdf>