MEAN-SHIFT CLUSTERING FOR INTERACTIVE MULTISPECTRAL IMAGE ANALYSIS

Johannes Jordan, Elli Angelopoulou

Pattern Recognition Lab, University of Erlangen-Nuremberg, Germany
johannes.jordan@cs.fau.de, elli.angelopoulou@cs.fau.de

ABSTRACT

Mean shift clustering and its recent variants are a viable and popular image segmentation tool. In this paper we investigate mean shift segmentation on multispectral and hyperspectral images and propose three new algorithms. First, we improve segmentation performance by running mean shift on the spectral gradient. At the same time, we adapt a popular superpixel segmentation method to the multispectral domain using modified similarity measures from spectral mapping. Based on superpixels, we design two mean shift variants that both obtain competitive segmentation results in significantly reduced running time. For one variant, the speedup in our benchmark is over 100 times. This enables mean shift clustering in an interactive setting.

Index Terms— Multispectral imaging, Hyperspectral imaging, Image Segmentation, Clustering algorithms, Distance measurement

1. INTRODUCTION

The ever-increasing popularity of multispectral data, especially outside remote-sensing applications, gave rise to general-purpose visualization and analysis software [1, 2]. These tools allow interactive inspection of a multispectral or hyperspectral image without prior knowledge. Materials and reflectance effects in the scene can be explored without the use of application-dependent dimensionality reduction or pixel classification. A viable tool for such analysis is clustering of the data. In particular, the mean shift algorithm [3] is a well-understood and popular clustering method that is in theory applicable to high-dimensional data, such as multispectral pixel vectors. Mean shift is a mode-seeking algorithm that clusters the image by associating all pixels to a set of common modes. Mean shift clustering has a key advantage over most popular graph-based segmentation algorithms: The clustering can happen in a feature space that is agnostic to pixel coordinates. No topological clues are needed and a segment can consist of several disconnected areas within the image that share a high pixel-wise consistency. It has also the important property that no other prior information is needed, e.g. a desired number of clusters. These properties make mean shift a good choice for clustering of a multispectral image to aid the user in further exploration without prior knowledge.

A major problem of the original mean shift is speed. In general, mean shift operates in $O(N^2)$ time, N being the number of data points, or pixels. This can lead to computation times of several hours for a high-resolution image, let alone a multispectral one. In 2003, Georgescu et al. [4] introduced the fast adaptive mean shift (FAMS) algorithm which significantly reduces running times by employing locality-sensitive hashing (LSH). However, on-demand calculation within an interactive usage scenario is still not viable. Recent methods further reduce computational complexity. Median shift [5] exploits the LSH aspect of FAMS by taking statistical properties of the LSH as a cue for finding modes in the feature-space density. While it achieves
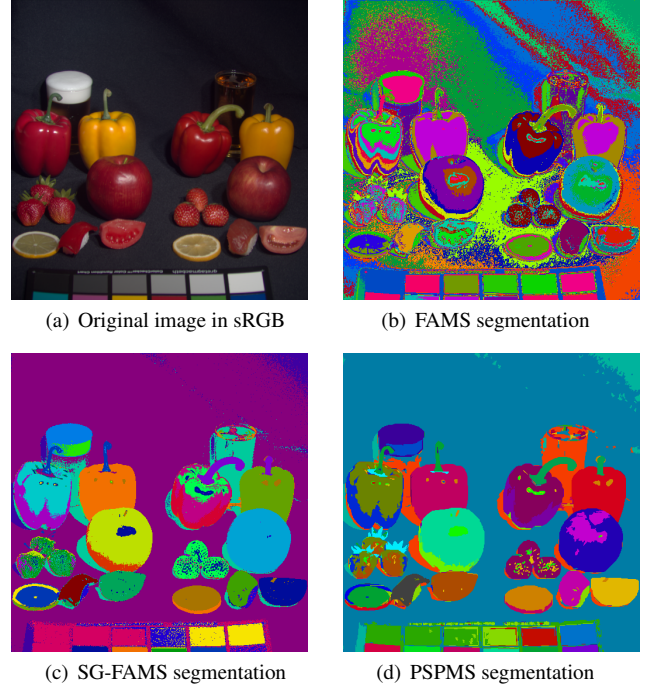

(a) Original image in sRGB   (b) FAMS segmentation

(c) SG-FAMS segmentation   (d) PSPMS segmentation

Fig. 1. 31-band *fake and real food* image [8] and segmentation results. We propose SG-FAMS and PSPMS. All figures best viewed in color.

a significant speed-up, it tends to under-segment the data: modes of less well-represented parts of the scene are not identified and these regions may be falsely attributed to poorly related larger segments. Variants of the so-called hierarchical mean shift [6, 7] do not tackle the theoretical computation time boundary. Rather, they start with smaller search kernels, which in practice reduces the number of data points to be considered in each shift step. Then, they iteratively restart the method with the obtained modes from the previous step as data points. A global segmentation is reached after two to four steps. However, reducing the kernel size becomes increasingly impractical with higher dimensionality of the input data. In a multispectral image, the feature space typically consists of 30 to 250 dimensions.

In this paper we provide a conclusive methodology for obtaining a valuable clustering of multispectral or hyperspectral images under tight time constraints. First, we show that desirable results can be obtained by FAMS when applied on the spectral gradient descriptor [9]. Second, we adapt an established superpixel segmentation method to multispectral data. Then, we define two novel variants of FAMS that employ superpixels for speed-up. We show that these methods provide competitive results to our multispectral FAMS variant with significantly decreased computational complexity.

2. MULTISPECTRAL FAST ADAPTIVE MEAN SHIFT

The FAMS [4] algorithm is a vital improvement to the general mean shift formulation for our application domain (high resolution images with a high dimensional feature space). We base our methods on FAMS and use its multispectral variant as a baseline as defined below.

2.1. Fast Adaptive Mean Shift

Mean shift is a density gradient estimator. It finds the modes of the multivariate distribution underlying a feature space with a kernel estimator. In our case, the feature space consists of the N spectral vectors $\mathbf{x}_i$ of length D. In FAMS, the mean shift vector at location $\mathbf{x}$ in feature space is defined as

$$\mathbf{m}_G(\mathbf{x}) = \frac{\sum_{i=1}^{N} \frac{1}{h_i^{D+2}} \mathbf{x}_i \, g\left(||\frac{\mathbf{x}-\mathbf{x}_i}{h_i}||^2\right)}{\sum_{i=1}^{N} \frac{1}{h_i^{D+2}} \, g\left(||\frac{\mathbf{x}-\mathbf{x}_i}{h_i}||^2\right)} - \mathbf{x} \,, \qquad (1)$$

where h_i is the bandwidth value as explained below, $G(\mathbf{x})$ is the Epanechnikov kernel, and g the kernel profile [4]. The algorithm consists of three steps:

Bandwidth Selection. Unlike the original mean shift which uses a fixed bandwidth h, FAMS selects an h_i for each data point determining its radius of influence. h_i is selected so that a minimum of k data points neighboring $\mathbf{x}_i$ lie within it. According to [4], parameter k is not critical to FAMS, however we need to consider it in Sec. 4.2.

Mean shift. $\mathbf{m}_G(\mathbf{x})$ is started at each data point and iteratively progresses in a hill climbing technique until it converges. The point of convergence is most often a mode (local maximum) of the density.

Mode Pruning. In the previous step we obtained a corresponding mode for each data point $\mathbf{x}_i$ originating from pixel P_i. In the pruning step, common modes of several pixels are identified and merged [3]. The segmentation consists of a cluster association C_i for each pixel, such that all pixels in a cluster share a common mode.

2.2. Spectral Gradient FAMS (SG-FAMS)

Our C++ implementation is based on MultiSFAMS [10]. We added support for higher-resolution images, parallelization of the adaptive bandwidth calculation, and Streaming SIMD Extensions (SSE) acceleration. We avoided parallelizing the shift step as it interferes with early trajectory termination. Figure 1(b) shows a segmentation obtained by applying our FAMS version on a 31-band multispectral image. At first glance, the result does not look satisfactory. However, further investigation reveals that the algorithm does in fact find distinguished clusters in the underlying distribution. The trouble is that geometric effects are very dominant (a problem not present in remote sensing data). The segmentation distinguishes brightness variations on objects of uniform material. Different materials however are often missed. For an example see both the plastic pepper on the left and the real pepper on the right in Figure 1(a, b).

To overcome this issue, we need to define a different feature space, similar to the use of the $L^*u^*v^*$ colorspace as opposed to RGB [3]. An appropriate representation for multispectral or hyperspectral data is the spectral gradient space [9]. The spectral gradient is the discrete approximation of spectral derivatives obtained by finite differencing. It is a descriptor that by design separates material information from reflectance content. Hence, in order to get a mean shift segmentation that focuses on material properties we use SG-FAMS, a spectral gradient variant of FAMS. In SG-FAMS we first compute the spectral gradient $\mathbf{g}_i$ for each pixel P_i, then use the $\mathbf{g}_i$ image as input to the FAMS algorithm.

Figure 1(c) shows an example segmentation obtained by SG-FAMS. We observe a small amount of segments that cover well the different objects, background and shadow regions. Different materials fall into the same segment only in rare cases (e.g. the missed strawberry leaves). Most specular highlights are assigned to a distinct segment. However, the necessary computation time does not fit into an interactive setting. Therefore we develop faster algorithms that employ superpixels.

3. SUPERPIXEL SEGMENTATION

We expect all pixels from a small, homogeneous image region to fall into the same FAMS cluster. Instead of finding a cluster for each pixel, we could search a cluster for such a small homogeneous region at once. Superpixel segmentation helps us to identify such regions. Superpixel is a term recently coined in the field of computer vision. It describes a set of pixels that are spatially connected and share high similarity, replacing the rigid structure of the pixel grid of an image.

3.1. Superpixel computation

A wide range of segmentation methods fall into the superpixel category, most prominently several graph-based algorithms as well as gradient-ascend methods. Achanta et al. [11] discuss and compare a comprehensive selection of these algorithms. While our method is agnostic to the superpixel segmentation method used, we chose the method proposed by Felzenszwalb and Huttenlocher in 2004 (FH04) [12]. FH04 has several advantageous properties that are particularly well-suited for our application. It is crucial for us that boundaries in the image are not missed and superpixels stay confined within a homogeneous region with high intra-similarity. The algorithm achieves the highest boundary recall in a benchmark of several superpixel methods [11] on the Berkely Dataset. Other superpixel methods often fulfill different properties, e.g. higher regularity in shape or size, that are of no concern to us. Also, FH04 has a time complexity of $O(N \log N)$. In the aforementioned benchmark, it ranks second in computation speed. This makes it an ideal choice for our interactive setting.

The algorithm operates on a 4-connected or 8-connected graph that represents each pixel as a node. Edges are weighted based on pixel dissimilarity. The graph is then partitioned starting with one partition, or superpixel, per node. Superpixels are iteratively merged while fulfilling several criteria. A parameter c manipulates the degree to which the difference between two superpixels must be greater than their internal differences to favor a split. Thus c influences the average superpixel size. See [12] for details. We configure FH04 to serve as a pre-processing step in clustering by setting a low c value.

3.2. Multispectral Superpixels

A key component of graph-based algorithms like FH04 is the setting of w_{ij}, the weight of each edge in the graph. In the original implementation, $w_{ij} = |I(P_i) - I(P_j)|$, where $I(P_i)$ is the intensity value of a grayscale pixel P_i. For RGB images, the authors run the algorithm on each band separately [12]. A weighted Euclidean distance was proposed for four band RGB+NIR images [13]. Both adaptations are not applicable to high-dimensional multispectral images.

A reasonable solution is to employ similarity measures specifically designed for spectral data. Two established and proven reliable spectral mapping measures are the Spectral Angle Mapper (SAM) and the Spectral Information Divergence (SID) [14, 15]. They best discriminate different materials based on their characteristic spectra.
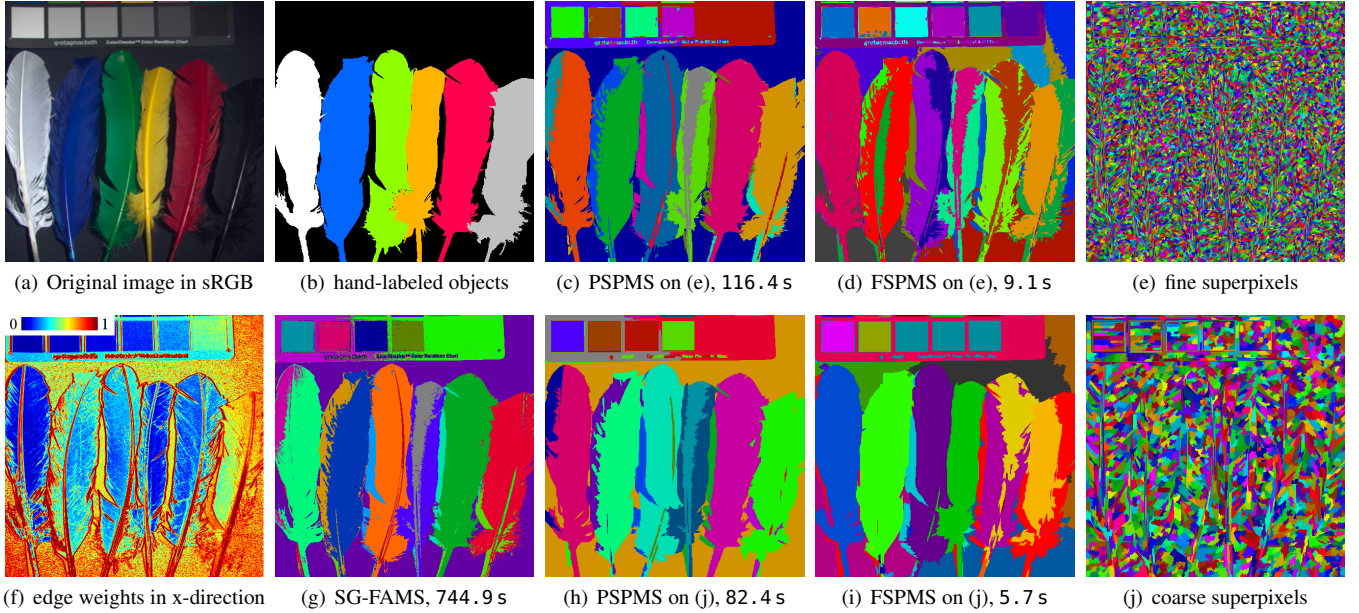
(a) Original image in sRGB (b) hand-labeled objects (c) PSPMS on (e), `116.4 s` (d) FSPMS on (e), `9.1 s` (e) fine superpixels

(f) edge weights in x-direction (g) SG-FAMS, `744.9 s` (h) PSPMS on (j), `82.4 s` (i) FSPMS on (j), `5.7 s` (j) coarse superpixels

Fig. 2. 31-band *feathers* image in sRGB, corresponding histogram-equalized SID edge weights, superpixel segmentations, and mean shift segmentation results compared to hand-labeled objects from [17]. Running times after image loading are denoted next to algorithm names.

SAM has already been used in conjunction with FH04 on remote sensing data [16]. Several similarity measures including SAM and SID were evaluated for another graph-based segmentation algorithm in [17]. The spectral angle (SA) has the property that it disregards pure intensity changes and is defined for two spectra $\mathbf{x}$ and $\mathbf{y}$ as

$$\mathrm{SA}(\mathbf{x}, \mathbf{y}) = \cos^{-1}\left(\frac{\langle \mathbf{x}, \mathbf{y}\rangle}{\|\mathbf{x}\|_2 \cdot \|\mathbf{y}\|_2}\right), \qquad (2)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product and $\|\cdot\|_2$ the L_2 norm. SID is based on the Kullback-Leibler information measure and models a spectral vector $\mathbf{x}$ of length D as a random variable $\mathbf{p}^{(\mathbf{x})}$ with $\mathbf{p}^{(\mathbf{x})}_{1 \le l \le D} = \frac{x_l}{\sum_{k=1}^{D} x_k}$. Then

$$\mathrm{SID}(\mathbf{x}, \mathbf{y}) = \sum_{l=1}^{D} \mathbf{p}^{(\mathbf{x})}_l \log \frac{\mathbf{p}^{(\mathbf{x})}_l}{\mathbf{p}^{(\mathbf{y})}_l} + \sum_{l=1}^{D} \mathbf{p}^{(\mathbf{y})}_l \log \frac{\mathbf{p}^{(\mathbf{y})}_l}{\mathbf{p}^{(\mathbf{x})}_l}. \qquad (3)$$

In several graph-based algorithms, e.g. graph-cut [18], only the order of weights is taken into account, but not their ratio. In our case, the ratio between weights is used when deciding whether to merge superpixels. We found that the highly non-uniform distribution of SID on our test images poses a problem. Therefore, we apply histogram equalization (with 10 000 bins for higher accuracy) on the edge weights. This procedure produces integer values, effectively reducing FH04 complexity[12]. Since histogram equalization improves the results for SA as well, it is always performed.

4. SUPERPIXEL MEAN SHIFT

We propose two new algorithms which combine multispectral superpixels with SG-FAMS. The idea behind both methods is to significantly reduce the amount of input data but maintain sufficient detail for obtaining a good segmentation quality. Depending on how this is done, considerable to vast speed-ups can be achieved.

4.1. Per Superpixel Mean Shift (PSPMS)

In this variant, we sacrifice spatial detail for execution speed. Superpixels $S_j, j \le N_S$ are computed on the original image. The feature space however is unchanged from SG-FAMS, i.e. it consists of the spectral gradient vector $\mathbf{g}_i$ of all pixels $P_i, i \le N$. For each S_j we compute the centroid $\mathbf{s}_j$ of all data points $\mathbf{g}_k, P_k \in S_j$. An adaptive bandwidth is selected for each pixel P_i. However, instead of starting the mean shift procedure at each $\mathbf{g}_i$, we start it at each $\mathbf{s}_j$. In all other aspects, PSPMS runs like SG-FAMS. As a last step, the cluster assignment C_j of S_j is back-projected to all $P_k \in S_j$. We obtain a full segmentation.

In our experiments we observe a fivefold to tenfold speed increase as compared to SG-FAMS. Algorithmic complexity is only reduced by a constant factor, as the adaptive bandwidth selection still has a complexity of $O(N^2)$. The mean shift procedure however is considerably more time consuming than bandwidth computation due to its iterative nature, and that is reduced to $O(N_S^2)$. Segmentation results of this variant best mimic SG-FAMS behavior, as the superpixels only affect the spatial resolution, but not the feature space.

4.2. Full Superpixel Mean Shift (FSPMS)

In this variant, we fully leverage the superpixel data representation. The centroids $\mathbf{s}_j, j \le N_S$ form the feature space. Adaptive bandwidths are computed per superpixel. Then, from each $\mathbf{s}_j$ the mean shift is performed. Cluster assignments are back-projected to the pixels after mode pruning. To achieve good results, we need to alter the algorithm in bandwidth selection and mean shift steps.

Bandwidth Selection. The bandwidth for each data point is chosen so that k nearest neighbors lie within its bounds. Georgescu et al. reason that the choice of k is not critical for the performance of the algorithm [4]. However, the adaptive bandwidths are directly related to k. If the feature space is sparsely populated, k has a strong effect on bandwidth size. As a result, data points influence a higher number of shift trajectories, leading to fewer distinct modes. While

Algo.	c	Superpixels	Segments	Seconds
SG-FAMS			$22.3 \pm\ \ 7.4$	629.3 ± 262
PSPMS	0.25	$2349 \pm\ \ 230$	35.0 ± 17.3	79.6 ± 32.8
	0.05	16706 ± 1178	33.0 ± 15.2	113.2 ± 39.9
FSPMS	0.25	$2349 \pm\ \ 230$	$23.7 \pm\ \ 8.2$	$\mathbf{5.9} \pm\ \ 1.1$
	0.05	16706 ± 1178	$27.5 \pm\ \ 6.8$	$9.0 \pm\ \ 1.5$

Table 1. Statistical results averaged over 13 test images.

typical input images have 2^{18} pixels (in our test images) or more, a superpixel segmentation on these images produces between 2^{11} and 2^{14} superpixels. Therefore a k suitable for SP-FAMS or PSPMS is not suitable anymore for FSPMS and would yield a broad under-segmentation. From our experiments we derive an effective rule for the choice of k: $k = p \cdot \sqrt{N}$. The linear factor p is a tunable parameter that influences the coarseness of the segmentation. Georgescu et al. also suggest to take the feature space dimensionality into account when choosing k.

Mean shift. The key idea behind the mean shift algorithm is the estimation of the density gradient. Our new feature space is based on the rationale that the superpixels provide a good sparse representation of the image's distribution. This is only the case when each superpixel both represents a homogeneous region in the image, and has the same weight as the points that are represented by it. Therefore, we weight the bandwidth of each superpixel centroid by the superpixel size.

The main advantage of this variant is that complexity is reduced to $O(N \log N + N_S^2)$. This makes the method viable for interactive applications, where a result should be obtained within a few seconds.

Figure 2 illustrates the operation and results of PSPMS and FSPMS on an example image. From the input image, displayed in sRGB, (a), SID edge weights are calculated including histogram equalization, (f). Superpixel segmentation is run with parameters $c = 0.05$, (e), and $c = 0.25$, (j). PSPMS and FSPMS are run with both superpixel configurations as input next to SG-FAMS.

5. EVALUATION

We test our algorithms on the CAVE multispectral image database [8]. This is a set of high-quality multispectral images that depict objects of different materials in a laboratory setting. Images have a spatial resolution of 512×512 pixels and cover the spectral range of $400\,nm - 700\,nm$ in 31 bands. Ground-truth data for evaluation of a global clustering is not available and would be very difficult to obtain. Therefore we mostly rely on qualitative evaluation of the segmentation results. An important measure in judging the segmentations is the number of obtained segments. In general, the segmentations of SG-FAMS provide enough detail without adverse over-segmentation. We expect PSPMS and FSPMS to obtain a similar number of segments as SG-FAMS, although in a fraction of time. A second important aspect is material discrimination. Objects of the same material properties should fall into one, or only few segments. Furthermore, objects that differ in material properties should not share a segment.

We tested both the SA and the SID similarity measures for edge weights w_{ij}. Both work very well when histogram equalization is applied. Most often, the segmentations obtained with SID were on-par or of higher quality. We, therefore, only report results computed with SID. We present two different settings of c to judge the influence of superpixel size. We found that considerably large superpixels can achieve good results in a shorter time. We keep the FAMS parameters fixed to $k = 1$, $K = 20$, $L = 10$ [4].
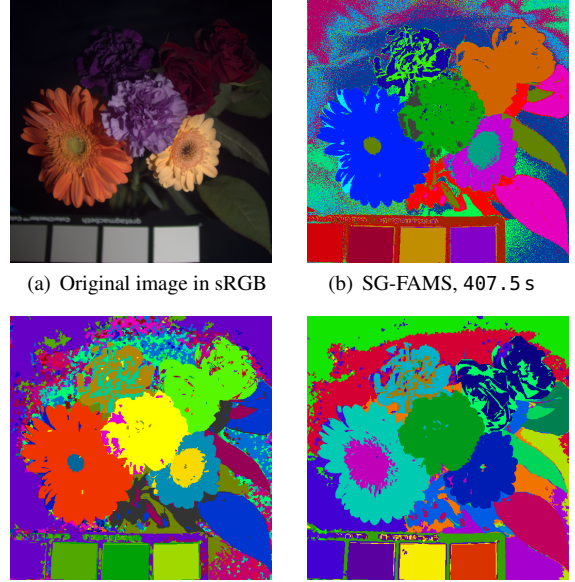

(a) Original image in sRGB   (b) SG-FAMS, 407.5 s

(c) PSPMS, $c = 0.05$, 120.0 s   (d) FSPMS, $c = 0.05$, 8.1 s

Fig. 3. 31-band *flowers* image and segmentation results.

Table 1 lists the number of obtained segments by each method, the number of superpixels used in PSPMS and FSPMS and the corresponding running times. All methods were run five times on an Intel Core i7-2600 CPU with eight threads. The number of segments is stable and in the same range across all methods. Example segmentations are depicted in Figures 1-3. A challenging example is the *flowers* image (Figure 3). A good segmentation of all six flowers is hindered by shading effects. PSPMS mostly differs from SG-FAMS in how it handles the noisy background. FSPMS puts emphasis on different aspects. All methods successfully capture the flowers as well as the leaves in the background.

We observe that on our data, PSPMS provides a reasonable speed-up without evident loss in segmentation quality. Another advantage of PSPMS over SG-FAMS is that it is more resistant to noise on the pixel level. FSPMS segmentations are not always on par with the other methods, as some details are missed in comparison. However, they prove functional and can easily be further refined in an interactive setting. The running times of FSPMS do offer an impressive advantage. In conclusion, FSPMS opens mean shift clustering to new time-critical application scenarios, including ours.

6. CONCLUSIONS

Unsupervised clustering of image data is an important image processing tool. Due to the rich pixel vectors of a multispectral image, mode-seeking clustering algorithms can provide very helpful segmentations without prior knowledge. We demonstrate this with the SG-FAMS method. Furthermore, superpixels are an effective tool not only on grayscale and RGB data, but also in the multispectral domain. By combining superpixels with SG-FAMS, we obtain a fast, yet reliable unsupervised clustering method. Our proposed FSPMS algorithm allows the on-demand calculation of a global segmentation in an interactive setting. The source code used in our experiments is available within the open-source *Gerbil* framework at http://gerbil.sf.net.

7. REFERENCES

[1] Johannes Jordan and Elli Angelopoulou, "Gerbil - A Novel Software Framework for Visualization and Analysis in the Multispectral Domain," *VMV 2010: Vision, Modeling and Visualization*, pp. 259–266, 2010.

[2] Björn Labitzke, Serkan Bayraktar, and Andreas Kolb, "Generic visual analysis for multi-and hyperspectral image data," *Data Mining and Knowledge Discovery*, pp. 1–29, 2012.

[3] Dorin Comaniciu and Peter Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[4] Bogdan Georgescu, Ilan Shimshoni, and Peter Meer, "Mean shift based clustering in high dimensions: A texture classification example," in *Computer Vision, 2003 Ninth IEEE International Conference on*, oct. 2003, vol. 1, pp. 456–463.

[5] Lior Shapira, Shai Avidan, and Ariel Shamir, "Mode-detection via median-shift," in *Computer Vision, 2009 IEEE 12th International Conference on*, sept. 2009, pp. 1909 –1916.

[6] Milan Šurkala, Karel Mozdřeň, Radovan Fusek, and Eduard Sojka, "Hierarchical blurring mean-shift," in *Advanced Concepts for Intelligent Vision Systems*, Jacques Blanc-Talon, Richard Kleihorst, Wilfried Philips, Dan Popescu, and Paul Scheunders, Eds., vol. 6915 of *Lecture Notes in Computer Science*, pp. 228–238. Springer Berlin Heidelberg, 2011.

[7] Milan Šurkala, Karel Mozdřeň, Radovan Fusek, and Eduard Sojka, "Hierarchical evolving mean-shift," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 1593–1596.

[8] Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K. Nayar, "Generalized Assorted Pixel Camera: Post-Capture Control of Resolution, Dynamic Range and Spectrum," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2241–2253, Sept. 2010.

[9] Elli Angelopoulou, Sang W. Lee, and Ruzena Bajcsy, "Spectral gradient: a material descriptor invariant to geometry and incident illumination," in *Computer Vision, 1999 Seventh IEEE International Conference on*, 1999, vol. 2, pp. 861–867.

[10] Elli Angelopoulou, Maithili Paranjape, and Tanuja Kothavade, "MultiSSA: MultiSpectral Scene Analysis Software," http://www5.cs.fau.de/fileadmin/Persons/AngelopoulouElli/MultiSSA, Jan 2012.

[11] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[12] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[13] Weihong Cui and Yi Zhang, "Graph based multispectral high resolution image segmentation," in *Multimedia Technology (ICMT), 2010 International Conference on*, oct. 2010, pp. 1 –5.

[14] Yingzi Du, Chein-I Chang, Hsuan Ren, Chein-Chi Chang, James O. Jensen, and Francis M. D'Amico, "New hyperspectral discrimination measure for spectral characterization," *Optical Engineering*, vol. 43, no. 8, pp. 1777–1786, Aug. 2004.

[15] Stefan A. Robila and Andrew Gershman, "Spectral matching accuracy in processing hyperspectral data," in *Signals, Circuits and Systems, 2005. ISSCS 2005. International Symposium on*. IEEE, 2005, vol. 1, pp. 163–166.

[16] David R. Thompson, Lukas Mandrake, Martha S. Gilmore, and Rebecca Castaño, "Superpixel endmember detection," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 11, pp. 4023–4033, Nov. 2010.

[17] Johannes Jordan and Elli Angelopoulou, "Supervised multispectral image segmentation with power watersheds," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 1585–1589.

[18] Camille Couprie, Leo Grady, Laurent Najman, and Hugues Talbot, "Power watershed: A unifying graph-based optimization framework," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 7, pp. 1384–1399, July 2011.