GraphSplatting: visualizing graphs as continuous fields

Robert van Liere Wim de Leeuw*

Center for Mathematics and Computer Science
Amsterdam, the Netherlands

Abstract

This paper introduces GraphSplatting, a technique which transforms a graph into a two-dimensional scalar field. The scalar field can be rendered as a color coded map, a height field, or a set of contours. Splat fields allow for the visualization of arbitrarily large graphs without cluttering. They provide density information which can be used to determine the structure of the graph.

The construction, visualization and interaction with splat fields is discussed. Two applications illustrate to usage of GraphSplatting.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation I.3.6 [Computer Graphics]: Methodology and Techniques

Keywords: information visualization, graph visualization, interaction

1 Introduction

The main problem of information visualization is the mapping of complex, non-spatial, abstract data onto effective visual forms. Relational structures, consisting of a set of entities and relationships between those entities is an important class of such data. Such structures are commonly modeled as graphs: the entities are vertices, and the relationships are the edges. A large number of graph drawing algorithms have been developed that present the information in a graph effectively. Effective graph drawings help the user to understand the underlying abstract data.

Graph drawing algorithms consist of two steps: layout and rendering. Graph layout is concerned with the placement of vertices and edges, such that properties of the graph are spatially conveyed. Graph layout algorithms are based on many different principles and show different properties of the graph. For example, algorithms can minimize the number of edge crossings, or can place vertices such that structure of the graph can be made visible. Graph rendering is concerned with mapping vertex and edge positions onto graphical objects and rendering these objects on the display. Usually graph drawing algorithms use discrete graphical objects to represent vertices and edges. Points or icons are used for the vertices, while curves are used to represent edges.

A key issue in graph drawing is how to handle very large graphs. Cluttering due to the large number of objects to display is very hard to avoid. Above a number of vertices it becomes impossible to display all vertices and edges using discrete objects. It will be impossible to discern between individual graphical objects and, as a result, only parts of the graph can be displayed in a single image.

In this paper we introduce GraphSplatting, a technique that represents a graph as a 2D continuous scalar field. This scalar field is called a *splat field*. 2D scalar field visualization techniques are used to render the splat field; e.g. as height fields, contours, color maps, etc. The utility of splat fields is based on the assumption that the density of points is an meaningful characteristic the chosen layout of the graph. Splat fields are useful to rapidly gain a overview of the complete structure of the graph.

A splat field can be used in combination with other graph rendering methods. Splat field is useful for obtaining an overview of the data and, after zooming into a detail, they can be combined with discrete graphical objects. In addition, splat fields can be used in combination with a second scalar field.

The paper is organized as follows: first we discuss related work. In section 3 we give the details of the GraphSplatting technique. We discuss the construction of the scalar field, and show how this field can be visualized. In section 4 we discuss various techniques that allow a user to explore data using a splat field. We show how zooming into splat fields can be realized, combining splat fields with texture fields, and use graphics hardware for fast splat field construction. In section 5 we illustrate GraphSplatting with two applications. The first application is concerned with the analysis of structure in multidimensional feature spaces. The second application is the analysis of the citation index of IEEE Vis'XX papers. We show that GraphSplattingcan be used to identify topics in visualization.

2 Related work

Visualization of graph data is a well studied subject. Excellent surveys of the layout and rendering of large graphs in the context of information visualization can be found in [4, 3].

Landscape visualization techniques have been used for text document visualization. Wise et al. developed the ThemeScape technique, which conveys information about topics in text documents, [13]. ThemeScapes are abstract, three-dimensional landscapes of information that are constructed from document corpora which augment a 2D landscape of text with a height dimension showing the strength of a theme in a given region. Elevation depicts theme strength, while other features of the terrain map such as valleys and peaks represent detailed interrelationships among documents and their composite themes. The authors claim that ThemeScapes are useful to rapidly gain a summary of the complete document corpus.

The work reported in this paper was inspired by the research summarized above. In contrast to other application dependent techniques, GraphSplatting is general purpose and makes use of standard data visualization techniques for rendering and interaction.

We also address the problem of combining a splat field with a second scalar field. In this way, density information depicted by the splat field can be correlated with the information represented by the second scalar field. From a visualization point of view, this problem can be regarded as simultaneously rendering two scalar fields. Many researchers have addressed this problem. For example, Trumbo [9] and Robertson et al. [7] have used color schemes for bivariate mappings. Also, combinations of contours and color or combinations of height fields and color are possible. S. Smith et al. and Weigle et al. have used iconographic displays for the visualization of multidimensional data [8, 12]. Van Wijk explored the usage of spot noise to generate texture patterns to represent an un-

*{robertl | wimc}@cwi.nl

derlying flow field [11]. Our approach is to make use of the spectral properties of the splat field. A splat field is shown to be restricted to low frequencies, allowing us to use the high frequencies to encode a second scalar field as a texture. Krueger applied the same principles to generate realistic surface textures in the context of ray tracing [5]. Also, the approach is in line with recent work in pattern recognition, in which image models are defined based on perceptual properties [6]. The claim is that ”periodicity”, ”directionality”, and ”randomness”, are the three most important dimensions of human texture perception.

The zooming and hardware accelerated techniques are based on those for interactive exploration in 2D vector fields using spot noise textures, [2].

3 GraphSplatting

GraphSplatting is a technique that transforms a graph into a continuous field. A central assumption is that the density of vertices is an important characteristic of layout used the graph. Layout techniques such as spring mass and other edge length minimization techniques have this property. Splatting is used to project each vertex of the graph a onto a two-dimensional scalar field using a splatting function. Instead of showing the individual vertices, the continuous variation in density is shown. Each vertex contributes to the field with a two-dimensional Gaussian shaped basis function. The resulting field is obtained by adding all the contributions. This field is called the *splat field*.

Figure 1 illustrates the mapping primitive. The figure shows a cross section of the Gaussian splatting function. The width of the Gaussian (σ in figure 1) determines the ’smoothness’ of the splat field. A large value of σ will result in smoothing out the details of the graph. Using a small value for σ will result in more detail of the graph. In the limit case $\sigma = 0$, the splat field shows vertices as impulses. σ is a global parameter which can be changed by the user.

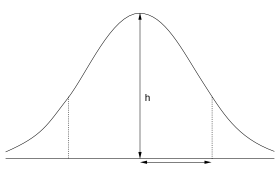

Figure 1: A cross section of the Gaussian splat.

GraphSplatting in itself is not intrinsically 2D, the equations for the splats could be easily extended to 3D. We choose 2D fields because the visualization of and interaction with 3D fields is more complicated. Also, there is no inherent three dimensionality in the data such as in 3D flow or medical volume data.

3.1 Splat field construction

In this section we define the construction of the two dimensional splat field more formally. The 2D continuous function M is constructed by summing the contributions of individual 2D basis functions:

$$M(\mathbf{x}) = \sum_{i=1}^{N} f_i(\mathbf{x}) \tag{1}$$

in which $\mathbf{x} = (x, y)$ is a position in the splat field. Each basis function is modeled as a normalized Gaussian function:

$$G^{\sigma}(\mathbf{x}) = \frac{1}{\sigma^2 \pi} e^{-|\mathbf{x}|^2/\sigma^2} \tag{2}$$

A property of the used Gaussian is that its integral is equal to 1 irrespective of the σ; i.e. $\int G^{\sigma}(\mathbf{x}) d\mathbf{x} = 1$. The basis function f_i is defined by placing the center of the Gaussian at the vertex position $\mathbf{p}_i = (p_{i,x}, p_{i,y})$:

$$f_i(\mathbf{x}) = G^{\sigma}(\mathbf{x} - \mathbf{p}_i) \tag{3}$$

The implementation of the splat field is done on a grid with a user controlled resolution. The Gaussian functions are discretized and added to the cells in the grid. The contribution of each splat to a pixel in the grid is estimated by using the distance between the vertex position and the center of the pixel. High resolution grids give a better representation of the splat field, however they take longer to compute. Section 4.3 discusses an implementation using graphics hardware acceleration.

Figure 2 illustrates how splatfields can be used. The left image shows 40 vertices of a graph (the edges are not shown). The middle and right image gives varying splatfield representations of the point collection. Both images use a different value of σ. The right image uses a larger σ resulting a smoother splat field. σ is a parameter providing the user a means to make a trade-off between a global overview of the structure of the data (right image) or more detail of the data (middle image). The overlapping of splats causes clusters of points to show up as a single maximum in the splat field. The mapping to gray value of the splat fields in both images is such that zero maps to white and the highest value in the field to black.

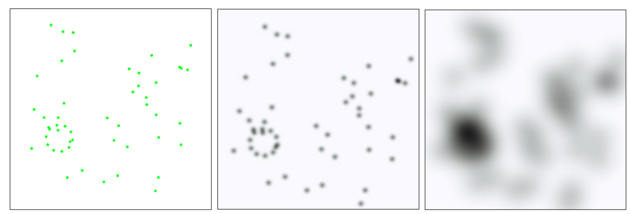

Figure 2: Visualization of a splat field. The left image show the layout of 40 vertices. The middle and right images shows two splatfields with varying σ values.

3.2 Splat field visualization

Standard visualization techniques can be used to visualize the function M. Although the GraphSplatting technique constructs a continuous scalar field, it is important to realize that the underlying data is a graph. As such, the splat field is often combined with discrete representations of the original data. These discrete representations can be rendered on top of the splat field.

Three standard data visualization techniques can be used to visualize the splat field :

- Color coding. A two dimensional view of the field in which the value is shown using color is useful for interaction. Different color coding can be used (for the black and white images in this paper the value was mapped to the gray level. For the color plates a color coding was used.). By combining the splat field with a discrete representation of the graph, interaction with individual vertices can be realized. Point-and-click of vertices to get additional information, dragging a vertex to a new position, etc, are interactions that are best performed in 2D.

- Height map. A three dimensional height map of the field is valuable to show the main structure of the field. A drawback of a height map is that interaction is difficult. For example, picking vertices or specification of a region is difficult in three dimensions.

- Isovalue contours. Contours can be used to show the boundary of specific clusters. A contour is also useful as a criterion to select all data items within the region bounded by the contour.

4 Interacting with Splat fields

4.1 Splat Field Zooming

For large data sets not all details can be discerned in the constructed splat field. This will be the case when even for very small values of σ adjacent splats will overlap.

However, the σ parameter in conjunction with a smaller region of interest does provide a natural interface to zooming. σ is automatically adjusted while the user zooms into a new region of interest. This is realized by maintaining a constant ratio between σ and the size of the region of interest during zooming; i.e.

$$\frac{\sigma}{S_{ROI}} = constant \tag{4}$$

in which S_{ROI} denotes the size of the region of interest. This ratio shows that when S_{ROI} is decreased, then σ decreases proportionally.

Figure 3 illustrates zooming. The splat field in left image gives an overview of an artificially generated data set. The left image show two clusters. The middle and right image illustrates zooming into regions of interest, which are indicated with a bounding box. The middle image shows that the cluster consists of three smaller clusters, and the right image shows the internal structure of one of the smaller clusters.

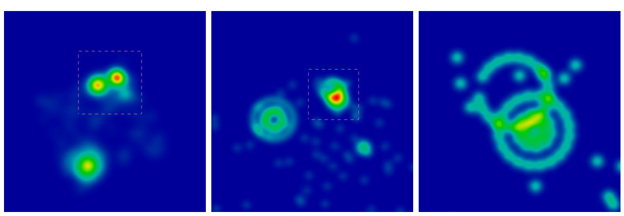

Figure 3: Splat field zooming. Left: splat field of all data. The region of interest is shown as a dashed bounding box. Middle: first level zooming. Right: second level zooming.

4.2 Combining Splat fields with Texture Fields

In section 3 the construction of a splat field as the sum of contributions of Gaussian bases functions (see equation 1) was discussed. A property of a Gaussian function is that it is limited in the frequency domain. The sum of Gaussian functions will have the same spectral properties as a single Gaussian function. As a consequence, the spectrum of the splat field is restricted to the lower frequencies. This property is used to map the additional scalar field. More specifically, the remaining higher frequencies are used for a texture that represents the additional scalar data.

There are many choices for the signal used to represent the additional scalar data. The choice we have made is to use high frequency noise of varying intensity which is added to the intensity of the individual splats. Although any signal can be used, regular signals may suffer from interference between overlapping splats.

The organization of the frequency domain is illustrated in figure 4. It shows how the frequency domain is organized into low

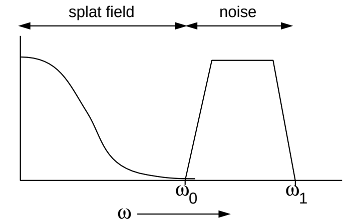

Figure 4: Organization of 1D frequency domain usage for splat fields and noise.

frequencies for the splat field and high frequencies for the noise. The low frequencies are bounded by ω_0. The range of the high frequencies available to map the second scalar field is $[\omega_0,\ \omega_1]$. ω_1 is a upper bound determined by the resolution of the signal representation.

We now discuss combining splat fields with additional scalar data more formally. The Fourier transform $g(\omega)$ of the normalized Gaussian (equation 2) is formulated as:

$$g(\omega) = \frac{1}{\sigma_k^2 \pi}\, e^{-|\omega|^2/\sigma_k^2} \tag{5}$$

in which $\sigma_k = \frac{1}{\pi\sigma}$ and ω is a 2D frequency vector. A large splat in the spatial domain (i.e. high value of σ) results in a low cut off in the frequency domain (i.e. low σ_k). Similarly, a small splat results in a high cut off in the frequency domain.

Combining splat fields with scalar data is performed by adding high frequency noise to the splat (see Figure 5). The intensity of the noise added to the splat is proportional to the mapped scalar value. The result is denoted as

$$S(\mathbf{x}) = (1\ +\ s_a R_h(\mathbf{x}))\ G(\mathbf{x}) \tag{6}$$

in which $R_h(\mathbf{x})$ is the high frequency noise function and s_a is a scalar attribute value of the point represented by the splat.

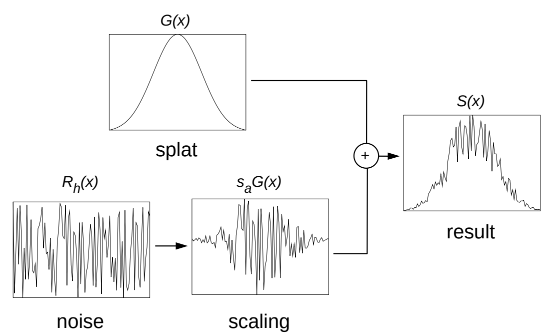

Figure 5: Noise added to splat function

A number of issues must be taken into account when implementing the method:

- The quantization of a signal results in a maximum representable frequency; denoted as ω_1 (figure 4). Graphics workstations have limited texture and screen resolution. As a consequence, the maximum frequency that can be represented by the workstation is determined by these resolutions. This maximum frequency fixes the value of ω_1 in figure 4.

- The user can control the splat width by changing σ, which fixes the ω_0 value. This results in the range $[\omega_0, \omega_1]$ which can be used for noise. If this range is too small, the noise mapping cannot be used. In our implementation noise mapping is not used if the value of σ is lower than 3 texels (=texture pixels).

- Instead of re-computing $S(\mathbf{x})$ for every rendered splat, our implementation uses a data base of 20 predefined splats for varying values of s_a. The splat with the closest s_a is chosen to be rendered. A data base of splats is useful to prevent jittering of texture when rendering sequences of splat fields and for increased rendering performance.

Consider figure 6 as an example of combining a splat field with a scalar field. The data set is a collection of points with two attributes. These attributes are the coordinates of positions on a regular grid. The left image shows the positions of the points. The middle image shows the splat field using a rainbow color map to denote the value of the field. Due to the regularity of the field, the splat field is almost constant at the chosen σ value. In the right image, the 'x'-attribute of the data points is linked to the described noise mapping. The result is a linear increase of noise intensity from left to right.

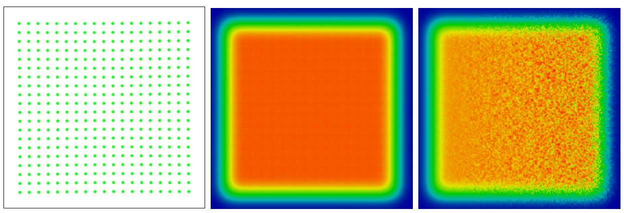

Figure 6: Three images of the synthetic regular grid data set. The left image shows the point positions. The middle image shows the splat field and the right image combines the splat field with a noise texture.

4.3 Hardware Implementation

Graphics hardware can be used to construct the splat field. This can be achieved by representing the Gaussian function as a textured polygon and rendering the polygon to an off-screen buffer with additive blending enabled.

Using the graphics hardware to construct a splat field introduces a problem caused by the limited depth of the frame buffer (often 8 bits per pixel). In an optimal rendering, the maximum intensity of the splat field would map onto the maximum pixel value. However, due to the additive blending of a unknown number of splats, the maximum intensity of a splat field is not known in advance.

We use an adaptive algorithm to achieve the near optimal intensity mapping. The idea of the algorithm is to scale the intensity of all splats such that the maximum intensity of the splat field is close to the maximum intensity of the frame buffer.

The algorithm is implemented by first rendering the splat field with an estimated splat intensity scaling. Then the maximum intensity I_{max} of the rendered splat field is determined by scanning the frame buffer. Figure 7 shows how I_{max} corresponds to the frame buffer intensities. There are three cases:

1. I_{max} is lies in region R2. Region R2 are those intensities that are less than the maximum intensity of the frame buffer FB_{max} and greater than 80% of FB_{max}. In this case, a satisfactory scaling is found and the algorithm terminates.

2. I_{max} is lies in region R1. Increase the splat intensity scaling with
$$S_{new} = S_{current} \frac{0.9\, FB_{max}}{I_{max}} \quad (7)$$
The value 0.9 is chosen such that the expected new maximum splat field intensity is in the middle of region R2. In this case, re-render the splat field with S_{new}.

3. I_{max} is equal to FB_{max}. Decrease the splat intensity scaling with
$$S_{new} = \frac{S_{current}}{2} \quad (8)$$
In this case, re-render the splat field with S_{new}.

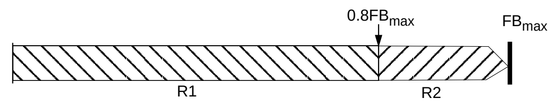

Figure 7: Relation between maximum splat field intensity and frame buffer intensities.

The desired scaling factor is found after a few iterations.

5 Applications

5.1 Multidimensional Feature Spaces

For any given image, a feature is expressed as a k-dimensional vector $f_i = \langle v_{i1}, v_{i2}, \ldots, v_{ik}\rangle$. For example, the brightness of an image maybe noted as a single value, i.e., $k = 1$, whereas a color histogram may consist for instance of 128 values, i.e., $k = 128$. Image similarity models often represent an image as a point in a multidimensional feature space where *similarity* of two images is expressed by the distance between their points in the feature space. A larger similarity/dissimilarity corresponds to a smaller/larger distance of the points.

Multidimensional scaling (MDS) can be been used to project multidimensional spaces onto a two-dimensional plane, [1]. MDS is a projection technique in which the discrepancy between the distance of two points in the multidimensional space and the projected plane is minimized.

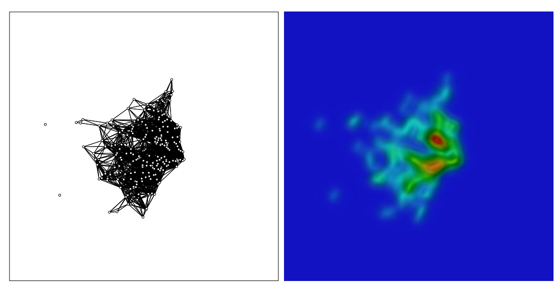

Figure 8: The projected image feature space represented as a graph (left) and a splat field (right).

We have used splat fields to interactively gain insight into the structure of feature spaces, [10]. Figure 8 shows two images of the projected multidimensional feature space. The left image of figure 8 shows a graph in which a vertex represents an image from an collection of images, and an edge represents similarity between two images based on the distance in the multidimensional feature space. The graph shows 200 vertices and 4007 (of the 200^2) edges. MDS has been used for the layout. The graph is very cluttered and it is very difficult to determine grouping of images.

The image on the right shows the splat field as a colored height field. By using the height map, the symmetry in the image collection with regard to the used features becomes apparent. Also, the location and relative sizes of the two main clusters of images can easily be detected. This information is difficult to obtain from the discrete rendering of the layout in the left image.

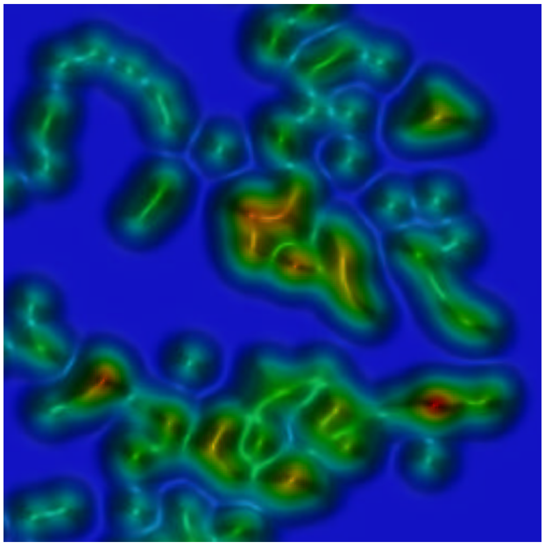

Figure 9: Zooming into an region of interest in the image feature space.

Figure 9 shows a detail of a high density area in the feature space. Now, the structures can be seen as a collection of individual points and similarity relationships in the vicinity of a particular point can be obtained. The figure shows many peaks in the height map that represent individual images, and a few larger peaks that represent small groups of very similar images. Zooming in still further would result in individual peaks for all images.

5.2 IEEE Vis Citation Index

We have applied GraphSplatting to the analysis of the IEEE Vis'XX citation index. The input data set are BibTeX entries of all papers in the proceedings of the IEEE Vis'XX conferences and all references between papers in this set. The data set consists 672 BibTeX entries and 1044 references. In addition to BibTeX information, the name of the session that the paper was presented in was collected; e.g. flow visualization related papers are usually presented in the session named "Vector Fields and Flow Visualization", etc.

A graph was defined in which vertices represent papers and references are represented as edges. A spring mass graph layout algorithm was applied to the graph, resulting in a layout in which referencing papers are attracted to each other, while papers that do not reference each other are repelled.

The goal of the visualization was to test the hypothesis that topics in visualization could be identified by using reference information. The motivation of this hypothesis is that papers about one topic often refer to other papers about the same topic.

The left image of figure 10 shows the output of the spring mass algorithm. As can be seen, aside from the papers which are not referenced and do not reference papers, there is a single component

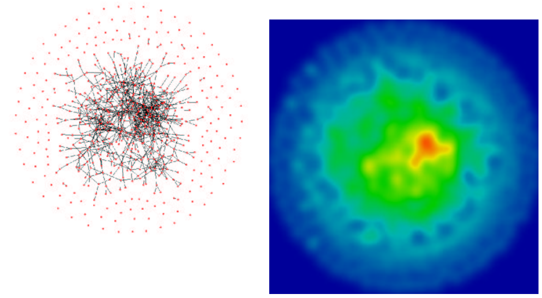

Figure 10: All papers published in IEEE Vis'XX conferences. The left image shows the graph. Small red discs represent papers and references between the papers are represented as lines. The right image shows the color coded splat field of the graph.

in the graph which can not be partitioned without breaking edges. The number of elements in the graph is too large and cluttered to get insight into the structure of the graph.

The right image of figure 10 shows the splat field applied to the graph. A rainbow colormap is used to show the values: blue denotes low densities, while red denote higher densities. The splat field was generated at a resolution of 512x512 and the splat size σ was set to 0.03 where the unit is the total field size.

According to the hypothesis, peaks can be interpreted as visualization topics. The higher peaks coincide with papers that are related to topics on which many papers are written (colored in red), while lower peaks (colored in green and yellow) denote less popular topics.

To validate the hypothesis, we used the session name as an indication of the topic to which a paper belongs. The noise mapping described in section refcombine was used to combine the splatmap with the information about the session name of the paper. The occurrence of the word 'flow' or 'volume' in the session name was linked as a boolean attribute to the noise mapping. In this way papers in the mentioned sessions were rendered as noisy splats while others were not. The left image of figure 11 uses the noise mapping to show papers that were presented in flow visualization sessions. The right image of figure 11 uses noise to show papers that were presented in volume visualization sessions.

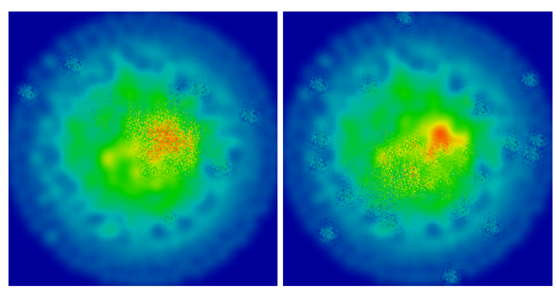

Figure 11: Combining the IEEE Vis'XX splat field with a texture field. The left image uses noise to highlight papers related to flow visualization while the right image uses noise to highlight papers related to volume visualization.

Two observations can be made from these visualizations. First, the images in figure 11 show that noise is limited to specific and mostly disjoint regions. In both cases, the noise field shows one large region and a number of smaller outliers. Second, the left image shows that there is a strong correlation between region of strong noise and the red region in the splat field. In the right image there is no maximum in the splat field in the region of strong noise.

From these observations we may deduce the following:

- Since the noise regions are disjoint, the assumption that visualization topics can be identified is valid.
- In the case of flow visualization papers, there is a strong correlation between the noise field and the splat field. For volume visualization papers, such a correlation is less apparent.
- The noise fields show small regions of outside the “primary region”. The reason for this is that the flow papers in outlying noise regions do not have references to other papers.

Figure 12 illustrates splat field zooming. The left image shows an overview of the splat field with a region of interest which roughly corresponds to the topic of flow visualization. The right image shows the region of interest using the zooming technique described in section 4.1. Using a smaller σ value shows that the large flow visualization region consists of three smaller regions. Manual inspection of papers in these regions revealed that the smaller regions roughly correspond to papers related to the sub-topics of flow texture, particle tracing, and flow systems.

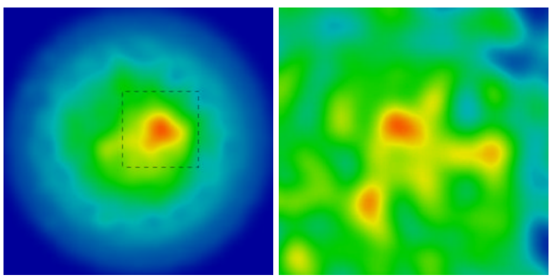

Figure 12: Zooming into an region of interest in the IEEE Visualization papers splat field.

6 Discussion

GraphSplattinghas two advantages compared to the discrete rendering of the graph: first an overview of arbitrarily large graphs can be produced. Since the splat field is constructed as an aggregation of basis functions, the number of vertices does not influence the amount of data to be visualized. Second, the visualization of splat fields presents density information, which is not easily available in discrete rendering.

The usage of splat fields is based on the assumption that the density of points, provided by the layout algorithm, is an important characteristic of the graph. The interpretation of the splat field depends on the interpretation of this density information. In the case of feature spaces, splat fields were used to gain insight into the structure of an image collection in the multidimensional feature space. Similarity relationships in the vicinity of a particular point in feature space can be obtained by studying the density of the points in the projected space. This information can used by feature developers to experiment with feature distributions and similarity models.

In the case of the citation index, a spring mass graph layout algorithm was applied to the graph, resulting in clustering of papers that refer to each other. The hypothesis is that papers in regions of high density form visualization topics. The splat field clearly showed the high density regions in the resulting layout. However, since no similarity information is available, additional techniques are needed to verify the hypothesis that the high density region were indeed visualization topics.

Noise mapping was used to present a secondary scalar field. Our experience is that the noise mapping technique, in contrast to other scalar representations such iconic mappings, scales to large graphs without cluttering.

7 Conclusion

GraphSplattingis a technique which transforms a graph into a two-dimensional scalar field. The scalar field is rendered as a color coded map, a height field, or a set of contours. Splat fields allow for the visualization of arbitrarily large graphs without cluttering.

Exploration is an important aspect of gaining insight in structural aspects of the graph. For this reason, the analysis of a graph using splat fields should be done interactively. An implementation utilizing graphics hardware and interactive zooming were developed. By varying the mapping parameters (σ and noise mapping), the user can change the information conveyed by the splat field without effecting the layout of the data. Because GraphSplattingworks on a fixed layout, the method does not depend on costly layout algorithms.

In this paper GraphSplattingwas applied to graphs, however, it can also be applied to any two dimensional point collection. For example it could be used show the sampling density for irregular sampled data.

References

[1] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1994.

[2] W.C. de Leeuw and J.J. van Wijk. Enhanced spot noise for vector field visualization. In G.M. Nielson and D. Silver, editors, *Proceedings IEEE Visualization’95*, pages 233–239. IEEE Computer Society Press, October 1995.

[3] G. di Battista, P. Eades, R.A. Tamassia, and J.G. Tollis. *Graph Drawing*. Prentice Hall, 1999.

[4] I. Herman, G. Melancon, and M.S. Marshell. Graph visualization and navigation in information visualization: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):1–20, 2000. (to appear).

[5] W. Krueger. Intensity fluctuations and natural texturing. In *Computer Graphics (SIGGRAPH ’88 Proceedings)*, volume 22, pages 213–220, August 1988.

[6] F. Liu and R. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(7):722–733, July 1996.

[7] P.K. Robertson and J.F. O’Callaghan. The generation of color sequences for univariate and bivariate mapping. *IEEE Computer Graphics and Applications*, 6(1):24–32, Feburary 1986.

[8] S. Smith, G. Grinstein, and R.D. Bergeron. Interactive data exploration with a supercomputer. In G.M Nielson and L. Rosenblum, editors, *Proceedings IEEE Visualization ’91*, pages 248–254. IEEE Computer Society Press, 1991.

[9] Bruce E. Trumbo. Theory for coloring bivariate statistical maps. *The American Statistician*, 35(4):220–226, 1981.

[10] Robert van Liere, Wim de Leeuw, and Florian Waas. Interactive visualization of multidimensional feature spaces. In D.S. Ebert and C.D. Shaw, editors, *Proceedings on New Paradigms for Information Visualization (NPIVM'00)*. IEEE Computer Society Press, 2000.

[11] J.J. van Wijk. Spot Noise – Texture Synthesis for Data Visualization. In *Computer Graphics (SIGGRAPH 91 Conference Proceedings)*, volume 25, pages 263–272, July 1991.

[12] Christopher Weigle, William G. Emigh, Geniva Liu, Russell M. Taylor, James T. Enns, and Christopher G. Healey. Oriented sliver textures: A technique for local value estimation of multiple scalar fields. In *Graphics Interface*, pages 163–170, May 2000.

[13] J.A. Wise, J.J. Thomas, K. Pennock, and D. Lantrip. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In S.K. Card, J.D. Mackinlay, and B. Shneiderman, editors, *Readings in Information Visualization*, pages 442–450. Morgan Kaufmann Publishers, 1999.