# LunarVis – Analytic Visualizations of Large Graphs*

Robert Görke, Marco Gaertler, and Dorothea Wagner

Universität Karlsruhe (TH), Germany
{rgoerke,gaertler,wagner}@ira.uka.de

**Abstract.** The analysis and the exploration of complex networks nowadays involves the identification of a multitude of analytic properties that have been ascertained to constitute crucial characteristics of networks. We propose a new layout paradigm for drawing large networks, with a focus on decompositional properties. The visualization is based on the general shape of an annulus and supports the immediate recognition of a large number of abstract features of the decomposition while drawing all elements. Our layouts offer remarkable readability of the decompositional connectivity and are capable of revealing subtle structural traits.

## 1 Introduction

Current research activities in computer science and physics aim at understanding the structural characteristics of large and complex networks such as the Internet [1,2], networks of protein interactions [3,4], social networks [5] and many others. A multitude of laws of evolution and scaling phenomena have been investigated [6,7], alongside studies on community structure, e.g. [8], and traditional network analyses [9]. Heavily relying on mathematical models and abstract characteristics, many of these techniques highly benefit from, or even depend on feasible advance information about structural properties of a network, in order to properly guide or find starting points for an analysis. Adequate visualization methods for complex networks are a crucial step towards such advance information. Furthermore, due to the diversity of such analyses, customized visualizations concentrating on user defined characteristics are required.

Along the lines of the more general issue in the field of information visualization, see e.g. [10], visualizations of large networks naturally suffer a trade-off between the level of detail and the visible amount of information. In other words, a detailed representation of a graph often antagonizes the immediate perceptibility of abstract analytic information. In this work we propose a layout paradigm that tackles the task of detailed analytic visualizations for large graphs. Our approach incorporates the strengths of abstract layouts, while individually placing all nodes and edges, i.e. without hiding away potentially crucial details. The

* The authors gratefully acknowledge financial support from the European Commission within FET Open Projects DELIS (contract no. 001907) and the DFG under grant WA 654/13-3.

S.-H. Hong, T. Nishizeki, and W. Quan (Eds.): GD 2007, LNCS 4875, pp. 352–364, 2007.

general underlying shape of the layout is a (partial) annulus. Subgraphs, defined by a decomposition, are then individually molded into annular segments. The annulus has been chosen for three primary reasons, first, it offers immediate readability of hierarchies and decompositional characteristics. Second, it allows for an insightful segment-internal layout, and third, it provides a large area for the drawing of edges, permitting the perception of segment connectivity at a glance, which is a major focus of many applications.

The technique works in three phases. In the first, abstract phase, a network decomposition determines the general shape of the layout, defining and arranging the drawing bounds of each annular segment. The second phase initializes the drawing of individual nodes and the third phase determines the final layout by means of sophisticated force-directed methods. Our paradigm offers many degrees of freedom that can incorporate any desired analytic property, allowing for well readable simultaneous visualizations of complementary properties. Simple user parameters tune the focus of our visualizations to either inter- or intra-segment characteristics, and furthermore permit a scalable trade-off between the overall quality and the required computational effort.

This paper is organized as follows. Sect. 2 sets our work into the context of related work. Then, after giving some definitions and notation in Sect. 3, we present our new layout paradigm in Sect. 4. An empirical study, using real world examples of the physical Internet, collaboration graphs and road networks, is given in Sect. 5. Finally, we conclude the paper in Sect. 6 with a brief summary.

## 2 Foundations and Previous Work

In the past, several layout techniques have been developed driven by the ambitious goal to properly visualize complex networks such as the Autonomous Systems (AS) network. Two important approaches are the landscape metaphor [11] and network fingerprinting [12], examples of which are shown in Fig. 1 and Fig. 2, respectively. Introduced by Baur et al., the former modifies a conventional layout

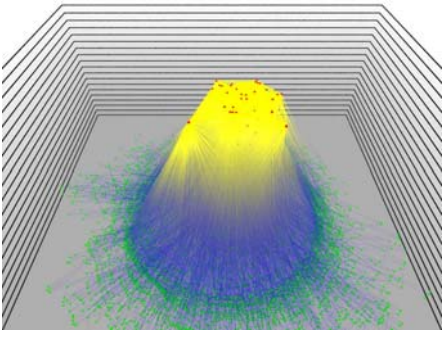

**Fig. 1.** A 2.5-dimensional layout (*landscape metaphor*) of the AS network [11]

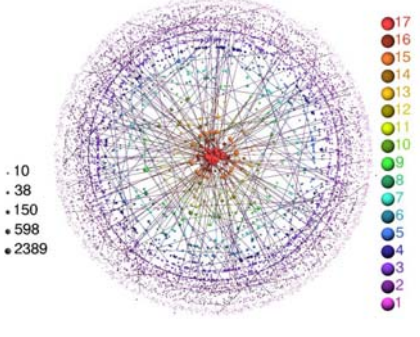


**Fig. 2.** A fingerprint of the AS network made with LaNet-vi [12]

technique by a framework of constraints that are based on analytic properties. The global shape of the network is induced by the position of structurally important elements, which automatically conceal inferior parts. Thus, it reflects the 'landscape' of importance, either in two or three dimensions. The latter approach, LaNet-vi [12] uses analytic properties to define the global shape, which consists of concentric rings of varying thickness, one for each level of the *core*-decomposition (see Sect. 3). Then, nodes are placed within these bounds and the overall readability is achieved by showing only a small sample of edges.

The above techniques and similar ones have succesfully been applied in numerous tasks, serving as an aide in network analyses. The method we present in the following synergizes assets of previous approaches and remedies a number of shortcomings in order to provide a layout technique that fingerprints a network (as LaNet-vi), but adds to this a much clearer visual realization of a number of analytic properties, thus offering a high informative potential. Before describing our visualization technique, we discuss the necessary preliminaries.

## 3 Preliminaries

Let $G = (V, E)$ be an undirected graph. We call a partition $P = \{P_0, \ldots, P_k\}$ of the set $V$ of nodes a *decomposition* with *shells* $P_i$. Furthermore, a *nested decomposition* $H$ is a nesting of subsets $V_i$ of $V$ such that $(V = V_0 \supseteq V_1 \supseteq \cdots \supseteq V_k \neq \emptyset)$. The sets $V_i$ of $H$ are called *layers*, giving rise to the *height* of $H$ being $k$ and the *height* of a node $v$, defined as the index $i$ such that $v \in V_i \setminus V_{i+1}$. The partition $P_H$ induced by a nested decomposition $H$ is canonically defined as $P_H = \{V_0 \setminus V_1, V_1 \setminus V_2, \ldots, V_k\}$. Edges between or within shells are canonically called *inter-* or *inter-shell* edges.

The choice of suitable network decompositions primarily depends on the field of application. In this work we focus on four different exemplary decompositions, *k-cores*, *clustering*, *reach* and *betweenness centrality*. The concept of *k-cores* was originally introduced in [13]. Stated in a procedural definition, the *i-core* of a graph is the unique subgraph obtained by iteratively removing all nodes of degree less than $i$, thus $k$-cores constitute a hierarchical decomposition. Graph clusterings commonly capture large scale inhomogeneities by grouping nodes into clusters by some formalization of the paradigm of *intra-cluster density* versus *inter-cluster sparsity*, see [9] for an overview. In the following we use a well known *modularity*-based graph clustering technique [8]. The betweenness centrality of a node is, roughly speaking, the number of shortest paths passing through it [14], reach is a similar concept used in transportation networking [15]. These decompositions are highly relevant in the analysis of large networks, such as protein network analyses [3] recommendation networks [8] and social sciences.

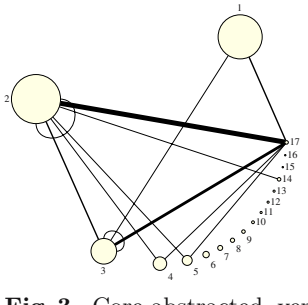


**Fig. 3.** Core-abstracted version of an AS graph

Visualizations of large networks usually suffer a trade-off between the details of visually shown elements and the amount of represented information. Widely known concepts resolving this are *abstraction*, as in Fig. 3 and the *reduction* of data to specific shells or parts of interest, illustrated in Fig. 4. While abstracted visualizations offer the best readability of these properties, much detail is lost, as in Fig. 3.

In contrast, zoomed visualizations as in Fig. 4 allow for the exploration of small scale subgraphs and structural subtleties. We overcome this compromise by using the layout of an abstracted graph as a blueprint but still draw all elements. Our goal is the visualization of all nodes and edges in a manner both pleasing and informative on intra shell characteristics, in addition to revealing the characteristics of the given hierarchical decomposition. We focus on properties like the size of shells and the connectivity within and between shells.

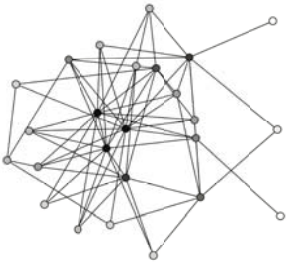

**Fig. 4.** Reduction of the 16-shell of Fig. 3

## 4 The Layout Technique

In the following we detail our construction technique for LunarVis. Roughly speaking, our approach divides up into three distinct phases, the first of which sets out the abstract layout attributes of the annular layout, such as the number of segments, their dimension and their placement. Based on these, a heuristic computation of suitable parameters follows, which will then be employed in the third and last step. This last, and by far the most intricate and computationally demanding step can be regarded as an iterative, segment-wise application of spring forces. These forces determine the final placement of each single node based on neighborhood attraction and repulsion both inside and between segments. In the end, we scale the annulus to the desired angular range and radial spreading and finally draw edges as straight lines with a high degree of transparency. Optionally, the size of a node and its color may serve as additional dimensions of information, yet ample use of these potentially overburdens a visualization. Algorithm 1 gives an overview of these three phases, which we describe in detail in the following sections.

### 4.1 Abstract Attributes

By any means, the informative potential of the our technique heavily relies on a suitable rough, abstract layout. We propose as the general underlying shape of the visualization an annulus, as shown in Fig. 5. The shells $s_i$ are lined up along a predefined angular range (here a full circle), placing the bottom ($s_1$) and the top shell ($s_8$) at the extremes. Thus, shells correspond to annular segments. User-defined properties then determine the individual dimensions of these segments, namely the *angular width* $\alpha_i$ and the *radial extent* $r_i$. In order to increase readability, small gaps $\beta_i$ that separate neighboring segments can be included.

**Algorithm 1.** LunarVis

**Input**: Graph $G = (V, E)$
**Output**: LunarVis Layout
1 Initialize abstract layout
2 Calculate parameters. Initialize random node placement within segments
3 **for** $i = 1, \ldots, \ell_{out}$ **do**
  **forall** *shells* $s$ **do**
    Project layout of $s$ to middle square $\overline{s}$
    **for** $k = 1, \ldots, \ell_{inter}$ **do**
      Apply inter-shell forces $\overline{s}$
    **for** $j = 1, \ldots, \ell_{intra}$ **do**
      Apply intra-shell forces to $\overline{s}$
    Project new layout of $\overline{s}$ to annular segment $s$
4 Finalize and scale annulus, draw transparent edges, color and resize nodes

The underlying annulus has an inner radius $r_{\text{in}}$ and an outer radius $r_{\text{out}}$, which, together with the angular range, define the total drawing area.

In our experiments, setting the annular segments to touch the inner rim and sizing them such that the largest shell also touches the outer rim, offered the best readability. For consistency, we let the number of nodes per shell define the angular width and the number of intra-shell edges define the radial extent throughout this paper, since these properties are generally of immediate interest. Molded into the underlying shape of annular segments, the shells can now be layouted individually. To give an impression of this step, and to point out the utility of an additional scaling function for the abstract layout, Fig.6 shows three layouts of the same network, using different scaling functions for the radial extent and the angular width of a shell. As canonic scaling functions, we used the strictly growing functions *square root* and *logarithm*. The network is a snapshot of the AS

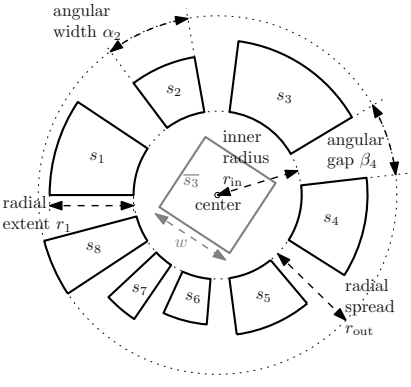


**Fig. 5.** The LunarVis annulus

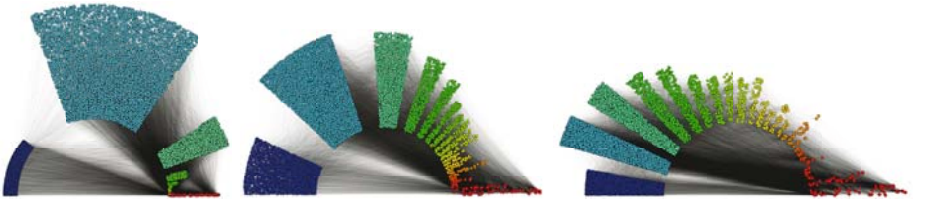

**Fig. 6.** Visualizations of the AS (1st March, 2005) using different scaling options. Radial/angular scaling is linear/linear (left), log/sqrt (middle), log/log (right).

network, decomposed into its core hierarchy. Individual nodes are left with a random placement, and the total angle is $\pi$. Linear scaling enables the immediate comparison of sizes, however, large values overshadow more subtle variations that do not become obvious without a logarithmic scaling of the radial extent. The inter-shell edge distribution is revealed by logarithmically scaling angular widths. Next, we describe how individual nodes are placed. For the sake of a better understanding we describe our parameter settings afterwards in Sect. 4.3.

### 4.2 Force-Directed Node Placement

Placing the individual nodes is by far the most computationally demanding task. Simple strategies offer an easy recognition of the shells' shapes, however, more sophisticated techniques can additionally reveal the internal structure of the shells while requiring more time and storage. Based on the forces proposed by Fruchterman and Reingold [16] we use spring- and repulsion forces to iteratively have the nodes of each shell adjust their position as suggested by their adjacencies and. In the following we describe this procedure in detail.

As sketched out in Alg. 1, our layout algorithm cycles through all shells a set number ($\ell_{\text{out}}$) of times by line 3. The nodes of a shell are then first subjected to inter-shell spring forces ($\ell_{\text{inter}}$ repetitions), thus moving towards their inter-shell adjacencies, and then, as a relaxational step, to intra-shell forces ($\ell_{\text{intra}}$ repetitions). To this end, we maintain a mapping of each shell, i.e. annular segment $s_i$, to a square $\overline{s_i}$ of size $w = 2/3 \cdot r_{\text{in}}$, centered at the origin and rotated such that it faces its original annular segment, see Figure 5. Forces are applied to the copies of nodes in the square $\overline{s_i}$, and then, the new coordinates of nodes in $\overline{s_i}$ are mapped back to the annular segment $s_i$ and its nodes are moved accordingly. Note that nodes in $s_i$ themselves exert inter-shell forces on their copies in $\overline{s_i}$.

Figures 7 and 8 illustrate the intention of this approach. First, note that a node coordinate $(x_{\overline{v}}, y_{\overline{v}})$ in a square shaped working copy $\overline{s_i}$ is obtained by transforming the circle coordinates $(\rho_v, \phi_v)$ in the annular segment $s_i$ in a canonical way, such that the angular position $\phi_v$ of $v$ within $s_i$ is linearly mapped to the the $x$-coordinate $x_{\overline{v}}$ within $\overline{s_i}$, and the radial position $\rho_v$ to $y_{\overline{v}}$. The rotation of $\overline{s_i}$ then aligns the $y$-axis of $\overline{s_i}$ with the middle axis ($\phi_{\text{mid}}$) of $s_i$.

The crucial idea behind this setup is that inter-shell forces pull nodes towards a specific side of the square, thus indicating their linkage tendency, while intra-shell forces relax

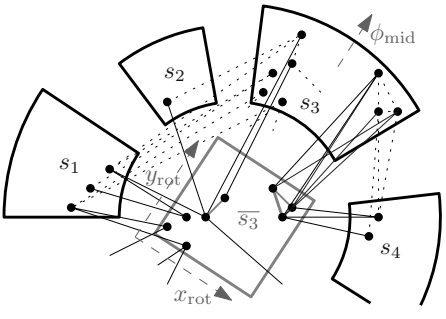


**Fig. 7.** Forces for $\overline{s_3}$ (excerpt). Inter-shell forces are caused by edges that link $\overline{s_3}$ with segments (solid, black). Intra-shell forces are attraction and repulsion of nodes within $\overline{s_3}$. Dotted edges are irrelevant during this stage.

the resulting crowding and unmask community structure and disconnected components. In Fig. 7, inter-shell forces draw the triangle of nodes in the right of $\overline{s_3}$ towards $s_3$ and $s_4$, while the nodes on the left, primarily being linked to other shells are pulled towards $s_1$, $s_2$ and other adjacencies. The subsequent application of intra-shell forces will keep the triangle grouped and separated, and relax the disconnected nodes on the left.

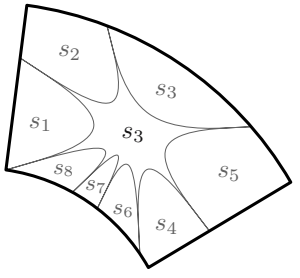


**Fig. 8.** Sketch of the preferred node locations in $s_3$

The areas of $s_3$ in Fig. 8 roughly sketch out where nodes, with a majority of adjacencies in shells as indicated, are drawn by inter-shell forces, before intra-shell forces relax the layout. The size and placement of these areas are induced by the abstract layout of the annular segments, see Fig. 5 for comparison. This segmentation of each shell allows for a sophisticated interpretation of a node's position.

Needless to say, we augmented our force-based algorithms with several well known techniques, such as soft clipping [16] to guarantee containment within shells, sentinel nodes that uncrowd segment borders [16] and an increased sluggishness of nodes with high degree [17]. However, (anti-)gravitational forces as well as simulated annealing [18], a randomized node ordering or an impulse history [17] yielded no substantial increase in quality, since our technique does not aim at a highly optimized local layout. We apply a simple exponential cooling, such that the movement of nodes is increasingly slowed. This proved necessary to avoid stubborn oscillations, especially if intra-shell forces are used purely relaxational.

An important observation is, that applying inter- and intra-shell forces at the same time naturally encourages force equilibria, but does not allow for a structurally targeted analysis. On the contrary, the separate application of inter- and intra-shell forces allows for a user-defined emphasis on either shell-internal properties or global connectivity.

### 4.3 Parameters

Heuristic or experimental assessment of parameters is inevitable when using customized force-directed methods. We base our forces on those proposed by Fruchterman and Reingold [16]. Alternative force models as proposed e.g. by Eades [19] or Frick et al. [17] did not prove more suitable but increased the running time, partly due to the fact that we do not enforce equilibria.

For intra-shell forces we set the base spring length to $C_i \cdot \sqrt{(\text{area}/\#\text{ vertices})}$, with the factor $C_i$ boosting the intra-shell spring length of dense shells. Depending on the decomposition, global factors for repulsion forces and spring lengths between 1 and 1.5 and 1.2 and 1.5, respectively, worked best. In fact, these two parameters were the only ones that required adjustment. Our inter-shell forces work with a base spring length of half the inner radius. Both the spring length and the spring force hardly needed additional tuning. Moreover, setting the edge

length $w$ of the squares $\overline{s_i}$ to significantly smaller values than $2/3 \cdot r_{\text{in}}$ blurred inter-shell forces, while much larger values exaggerated their range of effect.

As mentioned above, the iteration counters $\ell_{\text{out}}$, $\ell_{\text{inter}}$ and $\ell_{\text{intra}}$ are pure user parameters, since these govern the interaction and the emphasis of intra-shell and inter-shell aspects. In fact, surprisingly low iteration numbers already yields very nice results, a good starting point are $\ell_{\text{out}} = 10$, $\ell_{\text{inter}} = 10$, $\ell_{\text{intra}} = 5$. In the majority of drawings we used the logarithm for most scalings, as it copes best with power-law distributions and generally dampens overshadowing maxima.

## 5 Results

In the following, we present a selection of LunarVis visualizations, all offering many immediate insights. Nevertheless, knowledge about the drawing process, i.e. how nodes are placed, allows for a more structurally oriented interpretation.

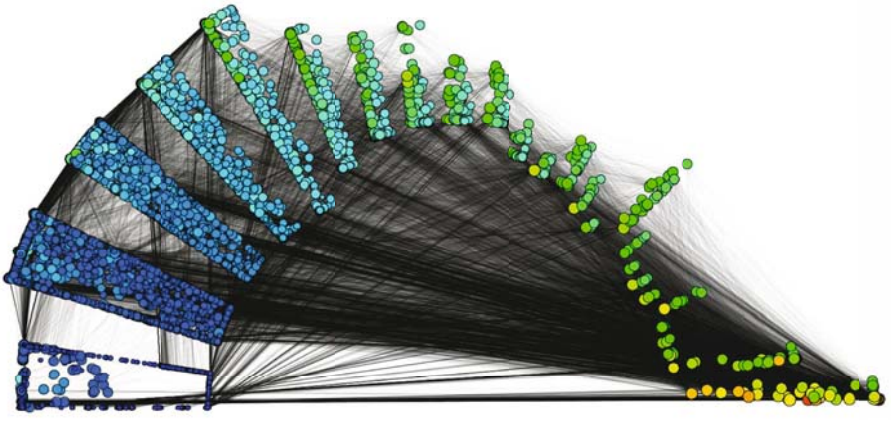

**Fig. 9.** A snapshot of the AS network taken at the 01.01.2006, decomposed by $k$-cores. Nodes with a high (low) degree are colored blue (red) and the area of a node is proportional to its betweenness centrality (all on a logarithmic scale). We chose a half circle for the total angular range and set the maximum shell at the right end.

Figure 9 reveals numerous characteristics of the core decomposition of the AS network at a glance. The well investigated fact that all shells primarily link to the core is immediate, alongside the observation that the internal communities of the first five shells are well interconnected (connectivity near outer rim), but not those of other shells. To name a few subtle facts visible in this drawing, note that mid-degree nodes can already be found in the 3-shell, that nodes with low betweenness are exclusively found in low shells while the opposite is not true, and that in low- to mid-shells nodes with higher degrees primarily link to lower shells, as they sit on the upper left. We used a time-sequence of such visualizations for an analysis of the temporal evolution of the AS network.

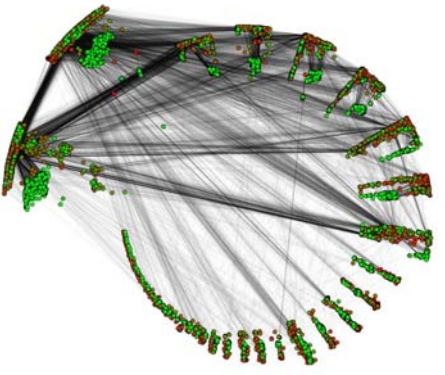

**Fig. 10.** The AS network, decomposed by a clustering. Nodes with a high (low) betweenness are colored red (green).

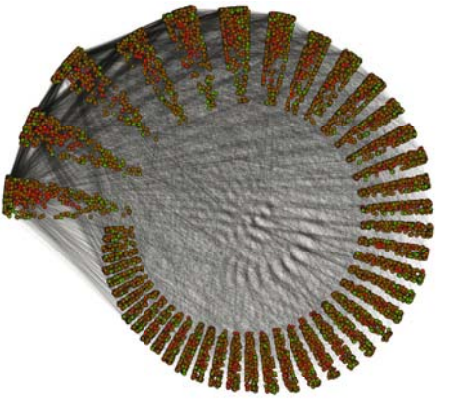

**Fig. 11.** A network created with BRITE [20], designed to emulate the AS. All parameters are as in Fig. 10.

For Fig. 10 and 11 a full annulus has been chosen due to the high number of shells (56 and 45). Figure 10 diplays the AS network, decomposed by community structure that has been identified by a greedy modularity based clustering algorithm [8]. The clusters are sorted by size. Figure 11 shows the same decomposition for a topology with the same number of nodes and edges, created with BRITE [20], an AS topology simulator. Quite clearly, BRITE fails to feature any of the peculiarities the AS network exhibits, such as high inhomogeneity in community sizes, the large number of tiny clusters or the fact, that most shells are almost exclusively connected to the two largest shells. An analysis yields clustering coefficients of 0.002 and 0.375 for BRITE and the AS network, respectively, and transitivities of 0.011 and 0.001, which agrees with these observations.

Figure 12 illustrates the core decomposition of an email network. The nodes represent computer scientists at Universität Karlsruhe, color coded by their department and sized by their betweenness, and edges are email contacts over the past eight months. As an exception, we used the sum of degrees for the radial extent with a square-root scaling for this LunarVis layout. From the multitude of observable features we point out the fact that community structure within departments is

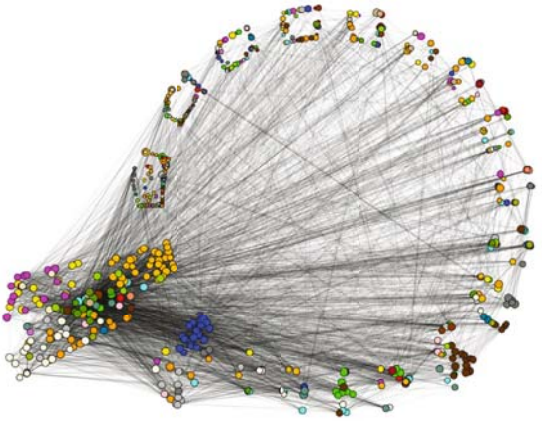

**Fig. 12.** Email network of the computer science department at Universität Karlsruhe

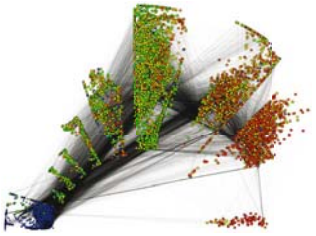

**Fig. 13.** Luxembourg roads, decomposed by betweenness, color indicates reach

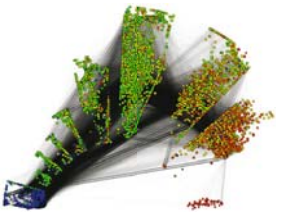

**Fig. 14.** München roads, decomposed by betweenness, color indicates reach

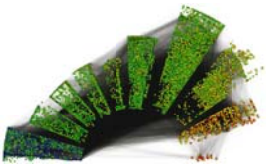

**Fig. 15.** European railroads, decomposed by betweenness, color indicates reach

**Fig. 16.** Luxembourg roads, decomposed by reach, color indicates betweenness

**Fig. 17.** München roads, decomposed by reach, color indicates betweenness

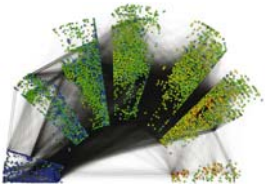

**Fig. 18.** European railroads, decomposed by reach, color indicates betweenness

been corroborated by the groupings in the top cores. As an example, the dark blue department, although being well interconnected (gathered), seems to have many contacts to lower shells, thus it sits at the inner rim of core 17.

Modern algorithms for route planning exploit numerous characteristics of road graphs for efficient shortest path computations, for an overview see e.g. [21]. Figures 13-17 display road maps of the Czech Republic and of the city of Munich, provided by PTV AG for scientific use, and Figures 15-18 display the European network of railway connections, provided by HAFAS. On the left hand side betweenness centrality [9], indexed into eleven logarithmically scaled intervals, served as the decomposition, and the figures on the right hand side are decomposed by reach centrality [15], colors are used vice versa. The stunning similarity of all corresponding drawings indicate that transportation networks share strong characteristics with respect to both reach and betweenness. However, several details can be observed that reflect intrinsic differences between these networks. Towards a taxonomy for transportation networks we can immediately observe that the railway network has very few hubs, both with respect to betweenness and reach. These are mainly capitals that, additionally, have exceptionally high degrees. The general correlation between reach and betweenness (color versus shell index) corroborates the fact that railroads constitute a scale-free network. This does not apply to either road network, which is due to the fact that road

networks tend not to have unique shortest paths – recall Munich's surrounding autobahn and Luxembourg's rural nature. The road networks strongly resemble each other, however, obvserve that in Munich, nodes of both maximum (autobahn segments) and minimum (residential dead-end streets) betweenness have a rather small degree. This cannot be observed in Luxembourg, where only nodes of minimum betweenness have an exceptionally small degree. From the facts revealed by the edge connectivity, note that hardly any peripheral nodes are adjacent to nodes of maximum centrality.

For computing our drawings, we used one core of an AMD Opteron 2218 processor clocked at 2.6 GHz, with 1 MB of L2 cache, running SUSE Linux 10.1. Our non-optimized development implementations in Java required drawing times between a few seconds and several hours, depending on the chosen number of iterations and the size of the network.

## 6 Conclusion

LunarVis is a new paradigm for drawing large graphs with a grand informative potential. Through sophisticated utilization of force directed drawing techniques and the neat design of an apt global shape, our technique creates visualizations of networks that reveal analytic properties of decompositions alongside properties of the shell connectivity at a glance, on the one hand, and offer insights into the interior characteristics of shells on the other hand. An emphasis on either inter- or intra-adjacencies can easily be adjusted.

The scope of application of LunarVis reaches far beyond mere network fingerprinting, as it does not only produce a distinct visual representation of a network but in fact offers the immediate recognition of analytic properties and unmasks structural characteristics and peculiarities. The transparent visualization of the set of inter-shell edges within the spacious interior of the annulus is particularly suitable for analyses on shell connectivity. LunarVis, however, is not a tool for investigating small-scale substructures or for purely esthetic, energy-minimal drawing.

Our results yield that LunarVis is highly feasible and informative in fields of application as diverse as internet studies, route planning and social sciences, employing decompositions by centrality, clustering and $k$-cores. The dimensions offered for analytic information surpass many existing visualization techniques in terms of perceptibility and detail, while the layout is highly configurable by using different scaling functions, emphasizing either intra- oder inter-shell relationships or simply plugging in complementary analytic properties.

The name of our paradigm *LunarVis* has been inspired by the semblance of our visualizations to the shape of the moon, sometimes waxing, sometimes full, but always a nice sight.

## Acknowledgements

We thank Ignacio Alvarez-Hamelin for profound discussions and brainstorming on LunarVis. Moreover, we thank Daniel Delling and Reinhard Bauer for

valuable discussions and data on transportation networks and Bastian Katz for helpful assistance in programming and documentation.

## References

1. Pastor-Satorras, R., Vespignani, A.: Evolution and Structure of the Internet: A Statistical Physics Approach. Cambridge University Press, NY, USA (2004)
2. Chen, Q., Chang, H., Govindan, R., Jamin, S.: The origin of power laws in internet topologies revisited. In: Proceedings of INFOCOM 2002, vol. 2, pp. 608–617. IEEE, Los Alamitos (2002)
3. Wuchty, S., Almaas, E.: Peeling the yeast protein network. Proteomics 5(2), 444–449 (2005)
4. Jeong, H., Mason, S.P., Barabási, A.L., Oltvai, Z.N.: Lethality and Centrality in Protein Networks. Nature 411 (2001) (brief communications)
5. Ducheneaut, N., Yee, N., Nickell, E., Moore, R.J.: "Alone Together?": Exploring the Social Dynamics of Massively Multiplayer Online Games. In: CHI 2006. Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 407–416. ACM Press, New York (2006)
6. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In: KDD 2005. Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 177–187. ACM Press, New York (2005)
7. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
8. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E 70 (066111) (2004)
9. Brandes, U., Erlebach, T. (eds.): Network Analysis. LNCS, vol. 3418. Springer, Heidelberg (2005)
10. Ware, C.: Information visualization: perception for design. Morgan Kaufmann Publishers Inc., San Francisco (2000)
11. Baur, M., Brandes, U., Gaertler, M., Wagner, D.: Drawing the AS Graph in 2.5 Dimensions. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 43–48. Springer, Heidelberg (2005)
12. Alvarez-Hamelin, J.I., Dall'Asta, L., Barrat, A., Vespignani, A.: Large scale networks fingerprinting and visualization using the k-core decomposition. In: NIPS (2005)
13. Seidman, S.B.: Network Structure and Minimum Degree. Social Networks 5, 269–287 (1983)
14. Freeman, L.C.: A Set of Measures of Centrality Based Upon Betweeness. Sociometry 40, 35–41 (1977)
15. Gutman, R.: Reach-based routing: A new approach to shortest path algorithms optimized for road networks. In: 6th Workshop on Algorithm Engineering and Experiments, pp. 100–111 (2004)
16. Fruchterman, T.M.J., Reingold, E.M.: Graph Drawing by Force-directed Placement. Software - Practice and Experience 21(11), 1129–1164 (1991)
17. Frick, A., Ludwig, A., Mehldau, H.: A fast adaptive layout algorithm for undirected graphs. In: Tamassia, R., Tollis, I(Y.) G. (eds.) GD 1994. LNCS, vol. 894, pp. 388–403. Springer, Heidelberg (1995)

18. Davidson, R., Harel, D.: Drawing graphs nicely using simulated annealing. ACM Trans. Graph. 15(4), 301–331 (1996)
19. Battista, G.D., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice-Hall, Englewood Cliffs (1999)
20. Medina, A., Lakhina, A., Matta, I., Byers, J.: BRITE: An Approach to Universal Topology Generation. In: Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Tele. (2001)
21. Wagner, D., Willhalm, T.: Speed-Up Techniques for Shortest-Path Computations. In: STACS. 24th International Symposium on Theoretical Aspects of Computer Science, pp. 23–36 (2007)