151

Parallel planning

An experimental study in spectral graph matching

Richard Schaffranek
Vienna University of Technology
richard.schaffranek@gmail.com

Abstract

A lot of attempts have been made in recent years to generate geometrically correct floor plans, spatial configurations and urban layouts in connection with functional relations and defined spatial properties(Elezkurtaj and Franck, 1999) (Duarte, 2001) (Donath, König and Petzold, 2012) (Nourian, Rezvani and Sariyildiz, 2013).Different heuristics (force based drawing...) / optimisations methods (metaheuristic solvers such as genetic algorithms, simulated annealing...)have been applied to search for the »best« trade of between a set of constraint/properties. Most of these techniques are based on an iterative and time-consuming process finding a good solution for one, two, maybe three different constraints/properties. But architecture is related to a multitude of different constraints/properties, which strongly depend on the given task and its context.

In image processing spectral graph matching has shown to solve different tasks such as graph drawing, image matching and image segmentation. A direct translation to architecture seems obvious as an image similar to a plan represents a particular spatial embedding of elements (pixels, rooms...) in two-dimensional space. As these problems are described through graphs, which represent the relation of elements rather than a particular spatial embedding, these approaches are applicable not only to a two-dimensional space but also in higher dimensions.

With such tools at hand, a prototype of a spatial configuration defined by a graph (functional or through properties such as e.g. integration, choice...) can be applied to an existing configuration (e.g. the refurbishment of existing buildings), to a spatial configuration which is generated by other properties than the prototype (e.g. structural, solar gain...) or to the most generic: A grid, resulting in a possible spatial embedding of the prototype. This allows to unlink functional / configuration lconstraint with other properties (an existing configuration, other configurational constraints/properties such as structural, solar...). Through unlinking, each constraint can be looked at on its own, developed on its own to a wanted solution and than relinked again. Further the process of matching two graphs, based on spectral graph theory, is not based on an iterative process but on one with a fixed computational time. Here the bottleneck is the calculation of the eigenvectors which can be preformed at least in $O(n^3)$.

Keywords

Generative planning, spectral graph theory, graph drawing, graph matching, parallel planning.

1. Introduction

There are only a few samples that apply spectral graph theory in the field of architecture and urban design. The work of Sean Hanna has to be mentioned: »Spectral Comparison of Large Urban Graphs«(Hanna, 2009), »Representation and Generation of Plans Using Graph Spectra« (Hanna, 2007) beyond that, literature in the field seems scarce. The later deals with the generation of floor plans, using a genetic algorithm, that shall match a given graph spectrum (eigenvalues). Hanna shows that this is a valid approach to create floor plans with similar local properties even so the overall size varies. However, it is still based on an iterative process and the floor plans (office layouts) generated lack certain basic functional constraint (e.g. accessibility to all areas).

This paper explores the use of spectral graph matching for the generation of floor plans and spatial layouts. As an input two different graphs are needed. One that represents an existing (Social to Spatial, Refurbishment), a generic (Prototype) or a generated spatial structure, an adjacency graph, and another that represents a relation that shall be matched onto the spatial structure (e.g. bubble diagram, social/work relation network). As these two graphs have to be provided the question how the proposed method is generative remains. The paper will show that a) the method can be applied in a generative way (Prototype), that b) the method can be applied to solve tasks often solved by e.g. genetic algorithms (Social to Spatial) and that c)the method enables two separate e.g. configurational and functional constraints / properties of a design task. This separation helps to generated solutions for the different constraints/properties efficient by e.g. the above-mentioned genetic algorithm.

Other than Hanna (2007)this paper will not only use the eigenvalues $\lambda_i$but also the eigenvectors E_i of a graph[1]. Technics developed for image processing and graph drawing will be examined on their applicability in the field of architecture. Therefore different matrix representations, methods of sign correction, matching and clustering will be looked at. The effects of different parameters (normalisation of the eigenvectors, limiting the number of eigenvectors) will be explored. To understand the impact of these parameters and to illustrate the potential usage within the filed, different samples will be discussed in detail.

As this paper focuses on the application of spectral graph matching in architecture and not on the mathematical background, the math behind will not be discussed in depth but sufficient references will be provided to guide the interested reader.

2. Graph drawing

In order to explain the idea and possibilities behind spectral graph drawing, the comparison of three different houses designed by Frank Lloyd Wright (March and Steadman, 1974) will be revisited.

»Sometimes, objects which appear to be very dissimilar on the first acquaintance may be seen, later, to share an underlying structural pattern. [...] Whilst they may look different, they are infact topologically equivalent. If each functional space is mapped onto a point [vertex V, editor's note] and if, when two spaces interconnect, a line [edge E, editor's note] is drawn between their representative points we produce a mapping known as graph. [$G=(V,E)$, editor’s note] « (March and Steadman, 1974: 28)

Such a topological mapping represented through an undirected graph is used to look for similarities. In the case of the first two examples used, the similarity is striking, however when comparing the first two examples with the last example, the similarity isn't obvious at first sight. (Figure 1, top row) In order to visually compare two graphs they would need to be drawn up in a similar way. For small graphs this can be done by hand (as March and Steadman have done in their book) but for larger graphs a methodical approach is needed.

[1] To compute the eigenvalues and eigenvectors the “Math.NET Numerics v3.2.0” library was used. http://numerics.mathdotnet.com/

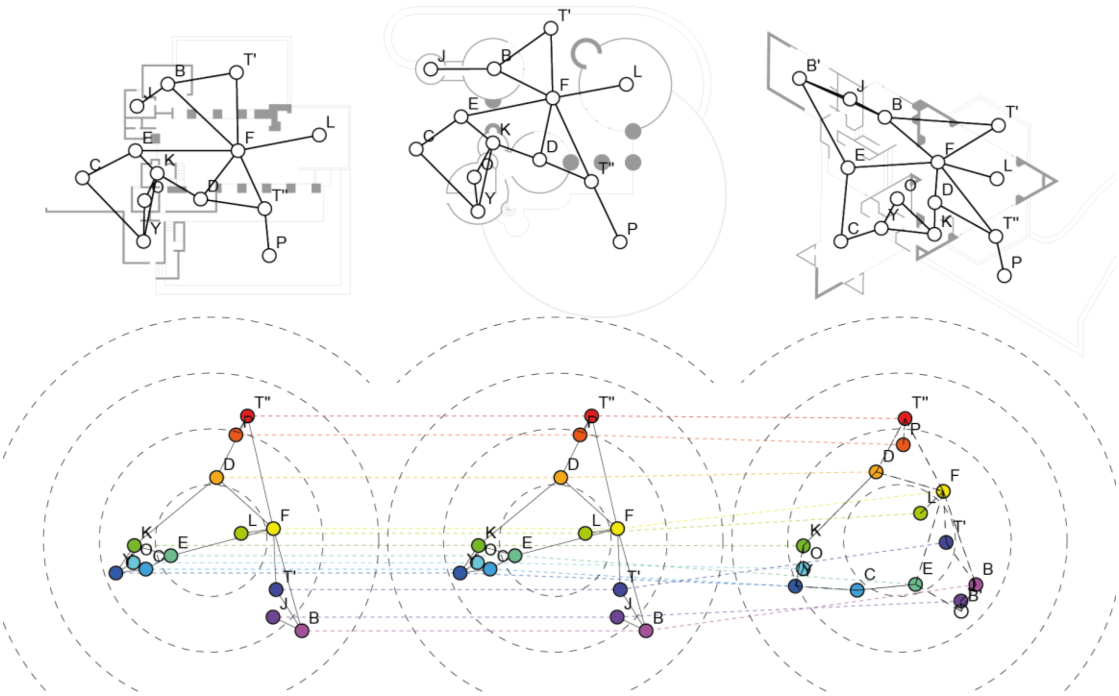

Figure 1: Three different houses design by Frank Llyod Wright (Little Private Club, Life Magazine, 1938; Ralph Jester House, Palos Verdes, California, 1938; Vigo Sundt House, near Madison, Wisconsin, 1941) and their adjacency graph as defined by March and Steadman (1974). For the Vigo Sundt House, the link between the kitchen (K) and the entrance (E) couldn't be verified and therefore the edge connecting K,E was removed in this paper. As well as the connection between the kitchen (K) and the living room (L) as it passes the dining area (D) (top row, left)
in the bottom row the 2D embedding of the graph using the 2nd and 3rd eigenvector of a normalized Laplacian matrix (Chung, 2006) can be seen. The matching between the second and third adjacency graph uses the first nine eigenvectors and applies a simple best first matching based on the distance matrix. The circles have a radius of 1.00, 0.75, 0.50, 0.25.

Spectral graph theory is applicable to the task of embedding any undirected graph into 2D space. This will not necessarily result in a planar representation of a graph but it will group the vertices of a graph in such a way that, nodes with a close relation within the graph will be situated close within 2D space. Therefore the eigenvalues and eigenvectors of the graph, represented bya $n*n$ matrixare calculated, where as n equals the number of vertices in the graph. The use of the Laplacian matrix L or the normalized-Laplacian matrix nL have proven beneficial as they ensure certain properties for the resulting eigenvalues and eigenvectors (Chung, 2006).

The number of eigenvalues, computed from the Laplacian matrix, that are equal to zero, equals the number of connected components of a graph. If there is only one connected component in the graph the values of the corresponding eigenvector will be equal, where as if there are more connected components within the graph the corresponding eigenvectors can be used to identify these components. (For more properties see Chung(2006)).

$$L_{ij} = \begin{cases} d(i) \text{ if } i=j \\ -1 \text{ if } (i,j) \in E \\ 0 \text{ otherwise} \end{cases} \qquad (0)$$

$$nL_{ij} = \begin{cases} 1 \text{ if } i=j \text{ and } d(j) \neq 0 \\ -\frac{1}{\sqrt{d(i)d(j)}} \text{ if } (i,j) \in E \\ 0 \text{ otherwise} \end{cases} \qquad (1)$$

In order to use the eigenvectors to embed a graph in 2D space, the eigenvectors have to be sorted according to the eigenvalues (in case of a normalized Laplacian matrix: $\lambda_0 \leq \lambda_1 \leq \lambda_2 ... \leq \lambda_n$). The rows of the first two eigenvectors corresponding to the smallest eigenvalues are used as feature vector (coordinates) for embedding the graph into two dimensions. Shawe-Taylor and Pisanski(1994) use the eigenvectors calculated form the Laplacian matrix, in this paperthe normalized Laplacian matrix is used instead, due to the better results achieved in graph matching.

The resulting embedding can be seen in figure 1. As the first two graphs have the same topological structure their eigenvalues are equal. The eigenvectors can differ in their direction. A step named signed correction has to be performed before matching. This is described in more detail in the chapter Exact Graph Matching. After sign correction the resulting feature vector for the embedding should be the same. The labelling of the graph (C, K, J…) plays no role in this process.

The embedding shows that spaces with a lot in common are placed closer to each other (e.g. B, J or Y,O,K,C,E).Others, which might share an edge but no further neighbours e.g. K and D are placed within a larger distance. As soon as a new edge is introduced e.g. between E and B'or others are missing e.g. between K and E this embedding changes (compare difference between Figure 1 middle and left example). New edges “pull” the vertices closer together, whereas the removal of an edge releases the vertices and lets them move apart, but the overall picture stays similar.

These observations are not only valid for two-dimensional feature vectors but apply as well to n-dimensional feature vectors, usable for graph matching.

3. Exact graph matching[2]

A task of matching two graphs, having the same amount of vertices has been demonstrated to be solvable with methods based on spectral graph theory. The idea is, if a graph has the same or a similar topological structure than the resulting eigenvalues and eigenvectors will be the same or similar and therefore the feature vectors of the graph. In Figure 1 this has been illustrated based on a 2D embedding of anormalized-Laplacian Matrix.[3]

A problem with eigenvectors, not so relevant for the drawing, but important when comparing two graphs, is that their direction (sign) is not unique. Shapiro and Brady (1992: 285) suggest a method for sign correction, maximally aligning one set of feature vectors to the other, which will be referred to as “minimizing costs” in this paper. Kosinov and Caelli(2002)suggest using the “dominant signage” test comparing the eigenvectors to maximally align the two sets of feature vectors.After this step of sign correction Shapiro and Brady (1992) suggest to calculate the Euclidian distance between all pairs of feature vectors between the two sets. Then they apply the Hungarian method[4] to compute the best match between the vertices of the two graphs.

In this case only the topological structure is considered. Matching the first two examples of figure 1 will result in a distance matrix where the matching feature vectors will have a distance of 0, since the topological structure and therefore the eigenvectors and the feature vectors are equal.

When trying to match one of the first two examples with the last example a few points have to be considered:

Firstly, the numbers of vertices differ between the two graphs (n, m), resulting in a different number of eigenvalues, eigenvectors and hence in a different dimensionality (n, m), of the feature vectors. To perform a matching the two set of feature vectors have to be of the same length. As pointed out

[2] In this paper the term exact graph matching indicates that both graphs have the same number of vertices on which a one-to-one matching is preformed.

[3] It is possible that two graphs, sharing no topological similarity might have the same eigenvalues and the same eigenvectors, which would result in a mismatch. However, it has been shown that the larger the graphs the more unlikely it is. The issue is been discussed in detail by van Dam and Haemers (2003).

[4] The Hungarian method solves the assignment problem. There are two sets of points. Each point from one set shall be assigned to a point within the other set so that the total cost of all assingments (e.g. distance between the points) is a minimum. The original algorithm was developed for two sets of equal size but it can be extended so that each point of the smaller set is assigned to a point of the larger set. (Compare:(Munkres, 1957))

by Shapiro and Brady (1992: 285-286) the ordering based on the eigenvalues already sorted the eigenvectors accordingly to their importance. So it is possible to simply truncate the feature vectors so that the dimensionality matchs. This can be done choosing the minimal dimensionality or any value smaller $k \leq min(m, n)$.

Secondly, the adaptation of the Hungarian method by Munkers (1957) could be used but a simply best first approach was chosen to find a best one-to-one matching in this case.

Figure 1 shows that a matching between the two graphs can be found (even so the graphs used here are more different than the graphs described by social/workrelation graphs, dark grey lines show strong connections, light grey lines showing week connections (right) with 26 vertices. The vertices in green are pre-matched.

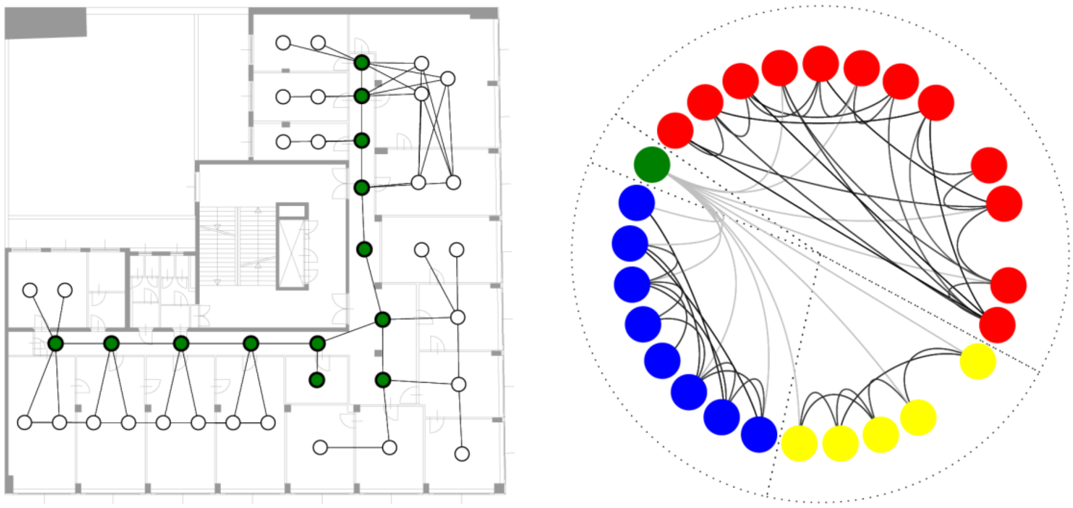

Figure 2: An existing floor plan and its adjacency graph A of desk spaces (left), with 41 vertices. A fictional

So far we have only looked at the topological structure of a graph, where two vertices are either connected or not connected through an edge. But in this case an edge contains more information represented through its weight w. To include this literature suggests different models: weighted adjacency matrix wA(Umeyama, 1988) weighted Laplacian matrix wL, Gaussian-weighted matrix wG (Shapiro and Brady, 1992).[5] When using an adjacency matrix or a Gaussian-weighted matrix the eigenvalues and their corresponding eigenvectors are, other than with Laplacian matrices, sorted descending $\lambda_0 \geq \lambda_1 \geq \lambda_2 \ldots \geq \lambda_n$

$$wA_{ij} = \begin{cases} w_{ij} \text{ if } (i,j) \in E \text{ and } i \neq j \\ 0 \text{ otherwise} \end{cases} \quad (2)$$

$$wL_{ij} = \begin{cases} \sum_{j=0}^{n} |w_{ij}| \text{ if } i = j \\ -w_{ij} \text{ if } (i,j) \in E \\ 0 \text{ otherwise} \end{cases} \quad (3)$$

$$wG_{ij} = \begin{cases} e^{-w_{ij}^2/2\sigma} \text{ if } (i,j) \in E \text{ and } i \neq j \\ 0 \text{ otherwise} \end{cases} \quad (4)$$

[5] List without any claim to completeness. Together with the Laplacian and the normalized-Laplacian matrix these are the matrix representations used within this paper.

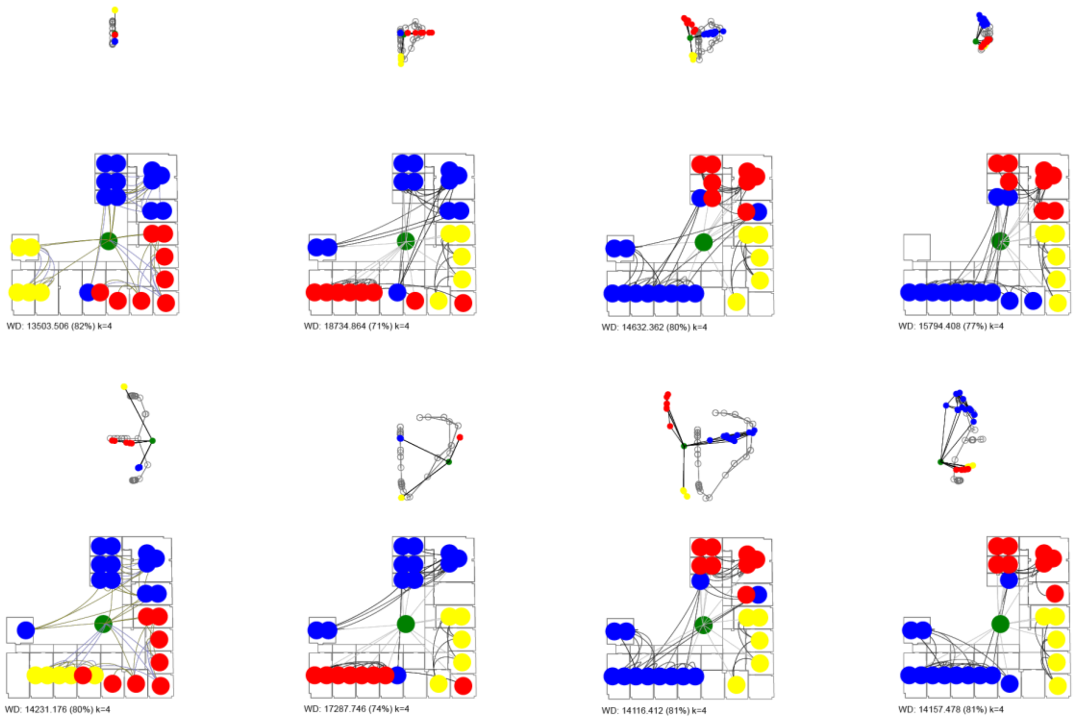

Figure 3: Matching of the social/work network-graph S onto the adjacency graph A with different parameters: matrix representations from left to right: weighted Laplacian matrix, weighted adjacency matrix, Gaussian-weighted matrix, normalized-Laplacian matrix; top row: feature vectors are truncated to $k=4$, bottom row: feature vectors truncated to $k=4$ and than renormalized (Equation 7).

When applying the same process as described before (Graph Drawing) to embed the social/workrelation network into 2D space, the groups (red, blue, yellow) which share a lot of edges with a large weight are placed close to each other, while the green vertex which shares edges with nearly all other vertices is placed near the centre of the embedding (Figure 3).

The second graph looked at is the adjacency graph A of the floor plan (Figure 2).A vertex can represent a desk or a part of the circulation. The edges are weighted based on multiplicative inverse ofthe Euclidian distance d between the vertices they connect.

$$w_{A:i,j} = \frac{1}{d_{i,j}} \qquad (5)$$

This is necessary, because as shown before, a large value will move vertices closer together whereas a smaller value or no edge lets vertices move further apart in the 2D embedding as well as in n-dimensions.

The aim is to find a one-to-one match between the social/workrelation network S and the adjacency graph A so that the sum of pathcosts $c_{A:i,j}$ between two matched vertices in A multiplied by the edge weight $w_{S:i,j}$ between those two vertices in S is minimized.

$$min: WD = \sum c_{A:i,j} * w_{S:i,j} \qquad (6)$$

In other words a one-to-one matching where the space allocation of the social/work network relations are optimized based on the idea of shortest paths. When following the steps as described for exact graph matching the resulting matches can be seen in figure 3. The best result is obtained when using the first four non-zero eigenvectors of a weighted Laplacian matrix. As discussed in

Conte et al. (2004) spectral graph theory offers an analytical approach to graph matching and can find a near optimal solution within polynomial time.[6]

To understand how good the found solution is a lower and an upper bound can be computed. A possible lower bound can be found by computing all path costs $c_{A:i,j}$ and sorting them ascending and the edge weights of the social/work network $w_{S:i,j}$ descending. Then multiplying the smallest path cost with the largest edge weight, the second smallest path cost with the second largest edge weight and so on until each edge weight is multiplied with a path cost. The sum of this sequence is the lower bound. To get the upper bound the path costs are sorted descending. These two bounds suggest that the quality of the solutions found, using different matrix representations, is within the range of 71 – 82%. For the bottom row results shown in figure 3 the feature vectors fV where renormalized (Equation 7)after truncation (This last step is suggested by Kosinov and Caelli (2002)). This seems to improve the overall quality of solutions where three out of four solutions are above 80%.

$$\bar{fV} = \frac{fV}{\|fV\|} \qquad (7)$$

As shown in this example a near to optimal solution for minimizing the overall path costs can be found applying spectral graph matching, this results in a low mean distance. Wineman et al.(2013) identified a low mean distance to fellow employees as spatial indicator for high innovation within an office environment. Whereas the other property identified by Wineman et al.(2013), high metric choice, could be used as bases for sign correction, aligning the eigenvectors in such a way, that more important feature vectors of S would be aligned to feature vectors of A with a higher metric choice.

4. Inexact graph matching

When the number of vertices diverge or when aiming at a one-to-many matching the above methods fail. Here Kosinov and Caelli (2002) have made avalid contribution to the topic.

Firstly, as mentioned before, they suggest renormalizing the feature vectors after truncation, with the argument that through such a step only the important properties such as the direction and orientation is used for matching and the other properties such as the magnitude are ignored. This is interesting in respect to the truncation of the feature vectors. If the eigenvectors are normalized also the feature vectors are of unit length hence the eigenvector matrix is orthonormal and the feature vectors are the rows of the eigenvector matrix. When truncating the feature vectors their magnitude will get smaller. The more values truncated the more influence on the magnitude.

Secondly, they suggest the use of clustering methods to preform the matching but do not specify which. For now, agglomerative clustering with different distance measurements (single linkage, complete Linkage, UPGMA, UPGMC, WARD, WPGMA)was used to find the matching.[7]

Thirdly, as suggested by Umeyama(1988) as simple hill climber (or other simple optimisation methods) can be applied to improve the result further. This comes with the downturn of introducing

[6]Other approaches might find better solutions but in exponential time $O(2^n)$. To illustrate this: The calculation of eigenvectors can be solved within $O(n^3)$ if $n=12$ then the algorithm would need $12^3=1728$, $13^3=2197$ steps an algorithm needing exponential time would need $2^{12}=4096$, $2^{13}=8192$ steps.

[7] Agglomerative Clustering starts to cluster a set of vectors based upon different distance measurement. In the first step all vectors are in separated clusters. The distance between each cluster is computed and the clusters with the smallest distance measurement are combined. This is continued until a predefined number of clusters is reached. In this case the number of clusters is defined by the smaller of the two graphs to be matched. Further the algorithm is altered so that the feature vectors of the smaller graph always remain in their unique cluster. An overview of the different distance measurement used, can be found under: http://de.wikipedia.org/wiki/Hierarchische_Clusteranalyse

an iterative process. However as the spectral graph matching gives a good initial match only a few steps by the hill climber are needed and therefore its application is reasonable in respect to computational time.

Refurbishment

One field inexact graph matching can be applied too, is the refurbishment of existing structures. Based on an existing or a possible spatial structure, which is generated through another process, an adjacency graph is computed. This can be seen in figure 4, where the subdivision of the floor is not based on the existing spatial structure but on a possible spatial structure taking only the loadbearing structure into account.

A simple bubble diagram shows the wanted spatial organisation including the needed infrastructure e.g. courtyards for daylight, staircases, toilets. The matching of these infrastructural elements is predefined. In the next step the matching based upon the method described in the chapter Inexact Graph Matching is computed (Figure 4 bottom left).The resulting matching is improved applying a simple hill climber resulting in the wanted layout (Figure 4 bottom right).

For this example the quality of a good matching cannot be defined by a distance measurement (Spatial to Social). Here a correct matching is achieved if the wanted bubble diagram is correctly represented in the spatial graph through the matching by: a) Connections represented within the bubble diagram shall exist connections not represented may exist. b) As the matching looks at a one-to-many matching of the graph all vertices of the spatial graph matched with a vertex of the bubble diagram have to be connected.

Figure 4 shows one of the best matching found. In order to understand all possible outcomes all different matrix representations suggested here in connection with the different distance measures for agglomerative clustering were tested. (see Figure 6, Appendix) This is possible since computational time is not a problem with the described method. The results are not conclusive for finding the best combination of matrix and clustering method but show that in two from 30 cases a correct matching was found. At first this seems like a poor outcome, especially since no correct matching was obtained before optimisation.

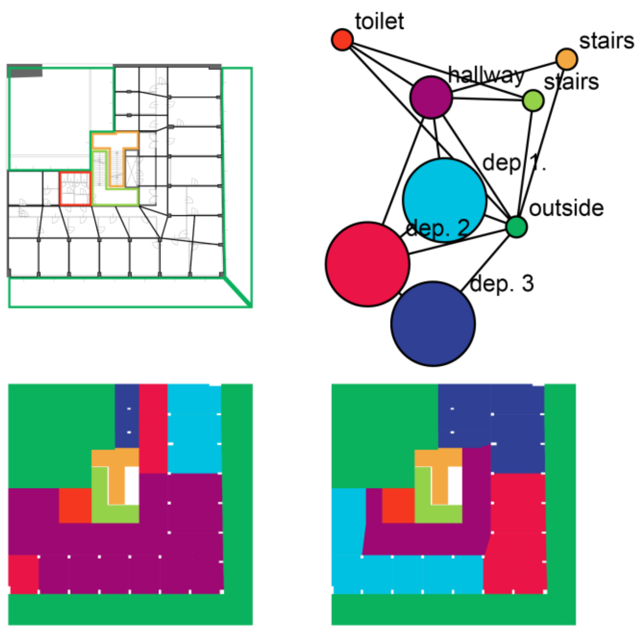

Figure 4: An existing floor plan with a spatial subdivision and pre-matches; Bubblediagram of functions (with relative sizes); The initial spectral matching (normalized-Laplacian matrix and UPGMC, $k=4$, Renormalized); Improved matching through a hill climber.

However, as described before a point to be considered is, that the direction of the eigenvectors is not defined, therefore their signs have to be corrected in such a way that two sets of eigenvectors maximally align. In this step the pre-matching is not accounted for, but should. This should be developed further. Taking a closer look at the results, in most cases only one/two connections are missing or an area is not connected, through customizing the last step of optimisation a further improvement can be made.

Prototype

The third example applies inexact graph matching, looking for a spatial configuration that would allow four flats, all with cross ventilation on one side of a staircase within three stories. If done in two dimensions (in one floor) the only configuration meeting the constraint of cross ventilation would result in one flat per side of the staircase (two flats / story). But when extending the grid in the third dimension a correct matching of the bubble diagram, modelling the constraint described above, with the adjacency graph of the 3D-grid can be found.

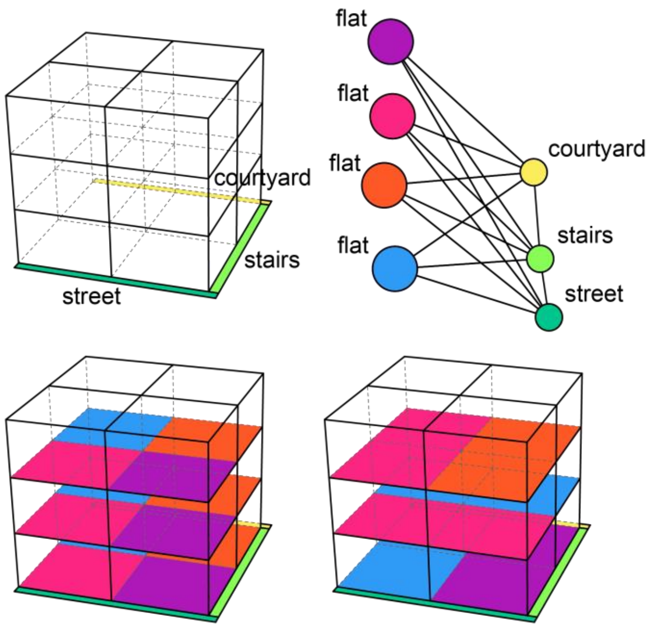

Figure 5: A 3D grid (5*5*3) and pre-matches (street facade, staircase, courtyard facade); Bubble diagram 4-flats, street facade, staircase, courtyard façade; The initial spectral matching (Laplacian matrix, WARD; Gaussian-weighted matrix, UPGMA or WARD;[illegible], Renormalized); Improved matching through a hill climber.

The given bubble diagram has only a small number of vertices, which can be grouped in three different groups in which the vertices share the same information to the rest of the network (group 1: flats, group 2: street, courtyard, group 3: stairs). This can be seen in Table 1, Laplacian matrix when looking at the Eigenvectors with the Eigenvalue of 7.

The eigenvalues and eigenvectors computed of the different matrix representations of the bubble diagram contain degenerated eigenvectors (their corresponding eigenvalues are equal).When including degenerated eigenvectors the feature vector of a vertex might depend on the sorting of the vertices.

Laplacian matrix

Λ	fV1	fV2	fV3	fV4	fV5	fV6	fV7
0	0.378	0.378	0.378	0.378	0.378	0.378	0.378
3	0.063	0.063	0.639	-0.764	0.0	0.0	0.0
3	-0.707	0.707	0.0	0.0	0.0	0.0	0.0
3	-0.496	-0.496	0.585	0.407	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.707	0.0	-0.707
7	-0.253	-0.253	-0.253	-0.253	0.077	0.856	0.077
7	0.208	0.208	0.208	0.208	-0.593	0.352	-0.593

Gaussian-weighted matrix

Λ	fV1	fV2	fV3	fV4	fV5	fV6	fV7
4.7407	0.311	0.311	0.311	0.311	0.421	0.509	0.421
1	0.402	-0.481	0.581	-0.502	-0.093	0.0	0.093
1	0.147	0.121	-0.241	-0.028	-0.673	0.0	0.673
1	0.640	0.286	-0.506	-0.420	0.195	0.0	-0.195
1	0.397	-0.650	-0.313	0.566	0.029	0.0	-0.029
-0.2761	-0.121	-0.121	-0.121	-0.121	-0.334	0.848	-0.334
-1.4646	0.373	0.373	0.373	0.373	-0.460	-0.148	-0.460

Table 1: Showing the eigenvalues and the corresponding eigenvectors of the bubble diagram given in Figure 5 represented by a Laplacian matrix and a Gaussian-weighted matrix.

Table 1 shows that only one eigenvalue other than zero is unique. The eigenvectors associated with the eigenvalue of 7 (Laplacian Matrix) would help to group the vertices but it is not possible to determine if the eigenvectors of degenerated eigenvalues can be used or not, even so it is clear in the given example. Other representations including the edge weights can be helpful e.g. a Gaussian-weighted matrix. Further Knossow et al.(2009) have suggested, instead of simply selecting the first k eigenvectors, to perform the sign correction and the selection of eigenvectors based on their histogram. This will be part of future work.

5. Future work

As shown spectral graph matching is an interesting field with a lot of possible applications in analysis and design that shall be explored in the future. The literature referenced here should give a good overview, but other publications that where not included might be of interest to address some of the described issues.

One of the issues is to align two sets of eigenvectors in the same direction, especially if vertices are pre-matched (Refurbishment, Prototype). Further methods of sign correction should be investigated e.g. histogram matching(Knossow et al., 2009) or a new method that can take other parameters (pre-matching, metric choice) into account should be developed.

In the last sample (Prototype) too many eigenvectors are degenerated and therefore not useable for matching, here future work should be done on how to overcome this issue. This might include other ways of modelling the graph or other matrix representations. Kunegis et al. (2010) demonstrate the use of signed Graphs enabling to include positive edge weights (attraction) and negative edge weights (repulsion) within the matrix representation. Or the representation suggested by Pisanski and Shawe-Taylor (1994), adding a value β to each edge weight, this can be imagined as an additional repulsion force between vertices not adjacent.

Other applications, beyond matching should be explored. The graph drawing process puts clusters that have a lot in common close, but it seems that also the global picture can help to understand the spatial configuration. The vertex F in figure 1 is placed close to the origin and seems to play a central role in the spatial configuration. Here it would be interesting to investigate the correlation between the distance and e.g. integration.

6. Parallel planning

The examples given seem dissimilar at first, beside the applied method of spectral graph matching but they all disconnect the spatial and functional side of a project. This gives the opportunity to focus on different aspects of a task independently as long as they can be represented as graphs and have a meaningful matching, hence the title: Parallel Planning.

Different to other methods such as metaheuristic solvers or force based graph drawing algorithms, the approach presented here is not depending on an iterative process and it has shown to be inventive in such a way that it can find solutions, which don't present themselves to the eye immediately (Prototype) and there for being generative. However it is based upon a very specific input, a graph, containing a lot of information. This information has to be generated upfront either by hand, reusing information developed in earlier projects or generated through the use of other methods e.g. the above mentioned metaheuristic solvers. As argued by Rittel and Webber (1973) architectural planning tasks are wicked and contain a multitude of constraint, changing for each project. Modelling all these constraint into a model seems impossible and further would need to be redone for each project. The concept of Parallel Planning would allow combining solutions generated with only one parameter with other solutions, resulting in a unique solution for a specific task.

References

Chung, F. (2006) *SPECTRAL GRAPH THEORY (revised and improved)*, [Online], Available: "http://www.math.ucsd.edu/~fan/research/revised.html" http://www.math.ucsd.edu/~fan/research/revised.html [09 Jan 2015].

Conte, D., Foggia, P., Sansone, C. and Vento, M. (2004) 'Thirty Years of Graph Matching in Pattern Recognition', *International Journal of Pattern Recognition and Artifical Intelligence*, pp. 265- 298.

Dalton, R.C. and Kirsanô, C. (2008) 'Small-graph matching and building genotypes', *Environment and Planning B: Planning and Design*, pp. 810 - 830.

Donath, D., König, R. and Petzold, F. (ed.) (2012) *KREMLAS: Entwicklung einer Kreativenevolutionären Entwurfsmethode für Layoutprobleme in Architektur und Städtebau*, Weimar: Verlag der Bauhaus-Universität Weimar.

Duarte, J.P. (2001) *Customizing mass housing : a discursive grammar for Siza's Malagueira houses*, [Online], Available: "http://dspace.mit.edu/handle/1721.1/8189" http://dspace.mit.edu/handle/1721.1/8189 [09 Jan 2015].

Elezkurtaj, T. and Franck, G. (1999) 'Algorithmic Support of Creative Architectural Design', Proceedings of the 17th Conference on Education in Computer Aided Architectural Design in Europe, Liverpool.

Hanna, S. (2007) 'Representation and Generation of Plans Using Graph Spectra', Proceedings of the 6th International Space Syntax Symposium, Istanbul.

Hanna, S. (2009) 'Spectral Comparison of Large Urban Graphs', Proceedings of the 7th International Space Syntax Symposium, Stockholm.

Knossow, D., Sharma, A., Mateus, D. and Houraud, R. (2009) 'Inexact Matching of Large and Sparse Graphs Using Laplacian Eigenvectors', *Graph-Based Representations in Pattern Recognition*, pp. 144-153.

Kosinov, S. and Caelli, T. (2002) 'Inexact Multisubgraph matching using Graph Eigenspace and Clustering Models', SSPR/SPR.

Kunegis, J., Schmidt, S., Lommatzsch, A., Lerner, J., De Luca, E.W. and Albayrak, S. (2010) 'Spectral Analysis of Signed Graphs for Clustering, Prediction and Visualization', Proceedings of the 2010 SIAM International Conference on Data Mining.

March, L. and Steadman, P. (1974) *The geometry of environment : an introduction to spatial organization in design*, London: Methuen.

Munkres, J. (1957) 'Algorithms for the assignment and transportation problems', *Journal of the Society for Industrial & Applied Mathematics*, March, pp. 32-38.

Nourian, P., Rezvani, S. and Sariyildiz, S. (2013) 'A Syntactic Design Methodology', Proceedings of 9th Space Syntax Symposium, Seoul.

Pisanski, T. and Shawe-Taylor, J. (1994) *Characterising Graph Drawing with Eigenvectors*, London.

Rittel, H.W. and Webber, M.M. (1973) 'Dilemmas in a general theory of planning', *Policy sciences*, pp. 155-169.

Rowland, T. and Weisstein, E.W. (2000) *"Orthonormal Basis." From MathWorld--A Wolfram Web Resource*, [Online] [13 Jan 2015].

Shapiro, L.S. and Brady, J.M. (1992) 'Feature-based correspondence: an eigenvector approach', *Image and vision computing*, pp. 283-288.

Umeyama, S. (1988) 'An Eigendecomposition Approach to Weighted Graph Matching Problems', *IEEE Transactiosn on PAttern Analysis and Machine Intelligence*, September, pp. 695-703.

van Dam, E.R. and Haemers, W.H. (2003) 'Which graphs are determined by their spectrum?', pp.241-272.

Weisstein, E.W. (2004) *Eigenvector -- form Wolfram Mathworld Web Resource*, [Online], Available: "http://mathworld.wolfram.com/Eigenvector.html" http://mathworld.wolfram.com/Eigenvector.html [13 Jan 2015].

Wineman, J.D., Kabo, F.W. and Davis, G.F. (2013) 'Spatial Layout, Social Networks and Innovation in Organizations', Proceedings of the Ninth International Space Syntax Symposium, Seoul: Sejong University.

Figure 6: Resulting spectral and optimized matches for the refurbishment example with specified graph; Different matrix representations and clustering methods; Sign correction: minimize Costs; $k=4$; Renormalized.

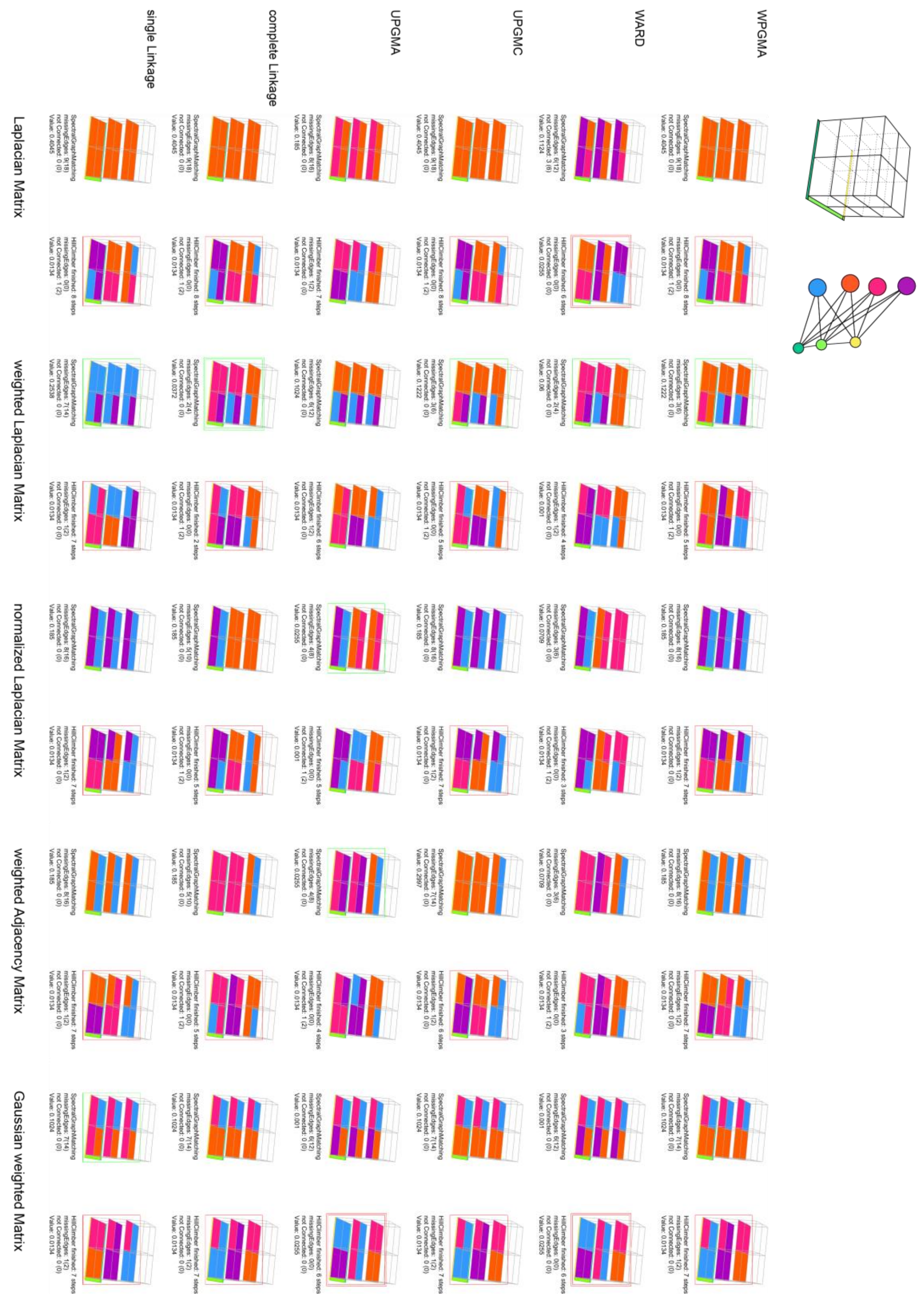

Figure 7: Resulting spectral and optimized matches for the prototype; Different matrix representations and clustering methods; Sign correction: minimize Costs; k=min(m,n); Renormalized.