

Contents

1	Introduction	1
1.1	Paper goal	1
1.2	MDS and LS	2
1.3	Papers on this topic	3
1.3.1	2021-03-29	4
1.4	Navigating in darkness	4
2	Theory	7
2.1	Mathematical Background	7
2.1.1	General Graph Theory	7
2.1.2	Angular Labels	8
2.2	Assessment of embeddings	9
2.2.1	Problem	9
2.2.2	Objective Functions	10
2.2.3	"Good" embeddings	11
2.3	Finding embeddings	12
2.3.1	Least square solution for Ω_{init}	12
2.3.2	Gradient descent approach	14
	How it works	14
	Gradients of objective functions	15
	Choosing initial parameters	15
2.4	Application for "Virtual Tuebingen"	16

2.5	Embedding with Normalization	16
2.6	Applying the Theory	16
2.7	Comparing different graphs	16
2.8	Comparing different optimization strategies	16
3	Practice	17
3.1	Survey design	17
3.1.1	Participants	17
3.1.2	Survey environment	17
3.1.3	Measurement of an angular direction estimation	18
3.1.4	Procedure	18
	Phase 1: Preparation	18
	Phase 2: North Estimation	21
	Phase 3: Measurement	21
	Phase 4: Travel	21
3.2	Data visualization	21
3.3	Data comparison	22
3.4	Conclusion	22
4	Discussion and Outlook	25
A	References	27
A	Supplementary data	29
A.1	Gradient computations	29
A.2	Survey measurements	29
A.3	Additional survey graphs	29

1 | Introduction

1.1 Paper goal

The overarching goal of this paper is to generate a procedure for the following problem:

We do have a map embedding or graph on which each edge does have an angular direction, e.g. edge $(\mathbf{x}_1, \mathbf{x}_2)$ does have an angle ϕ_{12} from node \mathbf{x}_1 to node \mathbf{x}_2 .

These edges can either be directed or undirected. When we have angles for an undirected graph we can assume an angle direction, e.g. $\phi_{12} \neq \phi_{21}$, although undirected edge $(\mathbf{x}_1, \mathbf{x}_2)$. With this knowledge from a given undirected graph we might just wanna generate a directed graph in which the angles in the other (not given) directions then are calculated by adding 180 or π . Or we want to jitter these calculated angles to make it more interesting. Our procedure is based on directed graphs, but with this assumption of generating directed graphs from undirected ones, we can generalize it to single-edge graphs. In future works weighted graphs might be assumed (see chapter 4)

Trees, also special cases of the general graph in graph theory, are a simple case for our procedure, as every directed tree has a perfect embedding minimizing angular (or other) errors. An undirected tree is, while a little bit more complicated, a simple case as well. A loop between two nodes does have an unique embedding (see example XY) for a given objective function, unique in the sense of relation of the two points with no rotation or translation assumed. From this follows that an undirected tree can be deformed to a directed one and does have a unique good embedding

We want to finding an embedding of the graph, namely a positioning of the nodes in a given space, in this paper the euclidean space \mathbb{R} . This embedding is by means of objective functions quantified, which are base on some aspect of error:

- Error in distance between embedding point and point generated from a node with its angle. Here are different distances for the generated point

possible, normalized or the same length as the embedding points are chosen here.

- Error in the angles, the one given as ϕ_{ij} and the other given from the embedding points themselves, namely the actual angle between the nodes calculated by means of the atan2-function for the embedding.
- Error in area given by the two vectors (embedding vector, generated vector by angle)

We want to find embeddings which do minimize specific objective functions and compare them (only visually?).

Minimization itself is done either by an analytical solution, e.g. the unique best embedding can be computed, or done by an optimization algorithm (which one?) which is a variant of the principle of gradient descent. These different approaches will be compared as well. The problem of the optimization process might be finding good parameters (start embedding, learning rate if necessary for the chosen algorithm) for every single graph again and again. This it way too cumbersome and might be worth investigating in following papers on this topic.

1.2 MDS and LS

Multidimensional scaling takes data and positions it in space. It positions the data in space in a way to minimize the difference between data and configuration. This is done to visualize (therefore most times in 2D or 3D space) similarities or distances while keeping as much information as possible within the configuration. Stress1, S-Stress, ALSCAL and the coefficient of determination are numbers quantifying the quality of a configuration generated by multidimensional scaling. This might be used here if we get to the point of using MDS for generating/generated embeddings.

The method of least squares on the other hand does a job of finding a function fitting the data in the most accurate way. The form or type of the function (logistic, line, squared) which underlies the data is hypothesized and will not be changed, only the parameters of this function are chosen to fit the data best.

We might be able to use the MDS approach for our purpose:

We define the distances as our angles between the pairs of nodes we have edges for. The problem is the fact, that we do not have angle data for every pair of nodes and we therefore aren't able to use the corresponding "dissimilarity matrix" or in our case the angle matrix \mathbf{A} or \mathfrak{A} . It seems that Matlab is able to calculate this problem with "NaNs" in the Matrix, e.g. no values or angles in our case.

TODO: This is what we will test for now and report back here in a bit!

1.3 Papers on this topic

In Ishikawa and Montello (2006) participants had to assess direction intuitions for ... At the end of this experiment four participants did draw a map, with the correlations with the real map given as well ... There was no visual representation of a graph generated from the given angle and distance estimates ... This graph representation will be produced for a similar study, where only angular estimates were conducted ... In Moar and Bower (1983) participants were to estimate angle directions of triads of streets, when imagining standing at one point, they were to draw an angular estimate for all 3 points in the triads ... They did generate estimates, which summed up to over 180, not possible for a map of a triangle in the euclidean space ... Again, here was given no embedding of these estimates, which will be the topic of this paper ...

In Yoder and Taube (2014) head direction (HD) cells are the topic ... These neurons fire when the animal's head (thus far no human experiments it seems) is directed in a certain angle ... The paper claims that cortical cells check information for the environment, while the subcortical ones check the angular head movements ...

In (Stevens & Coupe, 1978) topic is the problem with hierarchical components of spatial relations, e.g. when asked about towns in two different countries the initial estimate depends not on the position of the towns but of those of the countries ...

Our survey was conducted in a similar fashion to Ishikawa and Montello (2006) and Moar and Bower (1983) with the main difference, that participants were familiar with the surroundings ... Our goal was to embed the angular estimates in an optimal way ... This procedure can then be used for larger maps with embedding errors, for example Baumann (2019) ... Here, by correcting the angular error at each point (drift), we get an inconsistent map with errors ... A process of finding good embeddings for error prone map data with different optimization goals (error of distance, angles or areas) ...

The thematic of an egocentric pointing behaviour and thereof resulting odometry can be found in ...

Meilinger, Frankenstein, Simon, Bülthoff, and Bresciani (2016) interestingly states in his paper, that the recall of spatial information (example Tübingen, might work for survey) in form of positioning significant landmarks on a sheet of paper is dependend on the direction in which participants faced or where they were respective to the area they constructed (when facing south, south-up map; when east of the represented area, west-up map) As stated there, configuration maps were shown to be remembered north-up ...

Lancier paper, "However, distance estimates in spatial working memory are not derived from Euclidean space, but rather are assumed to be hyperbolically compressed. Whereas the place code estimates are assumed to be more veridical due to triangulation."

1.3.1 2021-03-29

apart from the SLAM algorithm, we can introduce the thought of different sensors applicable in the SLAM field (Zaffar, Mubariz, et al. "Sensors, slam and long-term autonomy: a review." 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS). IEEE, 2018.)

an overview of the SLAM field of study is given either in (Simultaneous Localization and Mapping: Part I BY HUGH DURRANT-WHYTE AND TIM BAILEY) or in (Khairuddin, Alif Ridzuan, Mohamad Shukor Talib, and Habibollah Haron. "Review on simultaneous localization and mapping (SLAM)." 2015 IEEE international conference on control system, computing and engineering (ICCSCE). IEEE, 2015.)

which compass direction is "up" in the cognitive map is irrelevant in our survey, as we conducted it and its data in such a way, that the participants did rotate, with the compass in hand, towards the questioned landmark (Meilinger, Tobias, et al. "Not all memories are the same: Situational context influences spatial recall within one's city of residency." Psychonomic bulletin review 23.1 (2016): 246-252.)

1.4 Navigating in darkness

As far as the field of robotics goes back in time, a prime goal has been not only to model and build robots and machines that can accomplish useful tasks that might be of too much work for a mere human to do, but to build and reconstruct the human person itself. From a biological or chemical perspective, there is no clear-cut distinction of external stimuli being received, the forwarding of this information as electrical impulses and its effect on an abundant number of neurons in the brain. Every 'step' in this information processing is in one form or the other done on a purely electrical basis. While perception and processing go wholly intertwined in the human brain, to model what is going on, we need to break down the vastly complex actions our body and brain are taking into small chunks ready to be analyzed and made sense of. This is true as well for the understanding of spatial representations of our surrounding world in our brain. How can we orientate our own person in surroundings familiar to us, if desired without help of our visual sense? Do we have integrated a map of named familiar surroundings embedded in another, bigger map of our whole ever perceived world? If so, how is the information of this map stored?

One probably has wandered around in the dark in his or her own house, myself for certain on multiple occasions. In most of these times I had no trouble finding the first step up or down the stairs or avoiding walking into the next wall head first, maybe while holding some beverage for an late hour working session. While pleasant, most of the time it's not worth mentioning or boasting of. Or, at least, so I tend to think. Admittedly, boasting should not be necessary, but mentioning and thinking about what underlies this capability of orientation on the other hand is quite interesting.

While one does not see its surroundings in the pitch black (let's assume with closed eyes, as otherwise the next light source out the window might shed enough light to grasp some visual information) this does not mean that same person is without sensory input about it. A scratch in a wooden floorboard, the ticking of some well-placed clock or even possibly the scent of previously brewed coffee on the kitchenette might give cues about the whereabouts of ones self. And cues as sensory input reach us all the time.

In this thesis, what is of interest here, is how we as humans do use cues of our surroundings to update our (spatial) representation of precisely these surroundings...

TODO: there is a long way to go still TODO: put all sources in bib! TODO: from Tristan over Slam and Mallot to this thesis...

2 | Theory

In this chapter we aspire to find a reliable way of calculating good embeddings of angle-labeled graphs.

The visualization and calculation of embeddings and the definition of itself presupposes a mathematical background, that will be established in section 2.1. The term "good embedding" is open to definition and will therefore be investigated in the subsequent section 2.2. With this fundamentals, the process of calculating embeddings can be stated and differentiated in sections 2.4 and 2.5. At the end of this chapter (section 2.6) we test our theoretical procedure of finding "good embeddings" for error-prone view-graphs. As this thesis takes up on the master's thesis of Baumann (2019), its final view-graph will be, besides procedurally generated graphs, subject to this test.

The overarching theoretical substance developed in this chapter will finally be applied for the analysis of a small survey in chapter 3. TODO: ausführung davon

2.1 Mathematical Background

2.1.1 General Graph Theory

In this section, let us establish the relevant subjects of *graph theory* we need for this thesis.

A graph is a mathematical construct of the form of an ordered pair

$$G = (V, E) \tag{2.1}$$

which consists of a set of vertices V and a set of edges E connecting vertices in a pairwise fashion. As we want to represent our graphs as directed, we define E as a subset of the cartesian product over the set of vertices V .

$$E \subseteq \{(v_1, v_2) | (v_1, v_2) \in V^2 \text{ and } v_1 \neq v_2\} \tag{2.2}$$

Let us denote here the number of vertices and edges in the graph as

$$\mathbf{n} := |V| \quad (2.3)$$

$$\mathbf{m} := |E| \quad (2.4)$$

An *embedding* of a graph does represent the mathematical structure of G in some space Σ . One example might be, that the embedding associates vertices $v \in V$ of G with points in the 2-dimensional Euclidean space \mathbb{R}^2 and edges $e \in E$ of G with straight lines or vectors connecting these points. As the view-graphs we want to analyse are basically top-down seen maps of the environment, these restrictions to Σ suffice.

Mark here, that any one graph can have an infinite number of different embeddings. The art of finding an aesthetic embedding for a given graph, from which one can easily deduce some of the graphs properties, is named the field and study of *graph drawing*.

For more detailed information not covered here see Bender and Williamson (2010); Diestel (2017).

2.1.2 Angular Labels

Taking up the work of Baumann (2019), we have to introduce the notion of angular directions to the graph and its vertices. These angles are assumed to be error-prone for the sake of real world examples.

As is the case in much literature of *simultaneous localization and mapping (SLAM)*, a robotic device, which is able to move around on its own term, is to map unfamiliar surroundings. (TODO: cite here some sources!) This robot, while moving around and dodging obstacles, is progressively trying to map his explored surroundings by means of measuring distances as well as angles between landmarks. (TODO: cite again) While the measurement is fairly accurate for distances, integrated lasers might suffice for that, the angles of rotations the robot actually performs are slightly off of its internally perceived measure thereof. These small measurement errors in angles are accumulating over the course of its explorations and can greatly disturb the actual map. The intrinsic map the robot builds, or in our case its underlying graph representation by means of angles, is therefore error-prone in regard to its own rotations.

Let us define a global reference direction $\nu = 0$, which can be seen as the objective "*north*" direction of a compass. In Baumann (2019) the accumulating angular deviation from the global reference direction is simulated and updated while the agent explores the environment. It is stored as label ν_i with a vertex v_i upon latter's finding in the mapping process.

Relative to ν_i we then are given the angular directions ϕ_{ij} from vertex v_i to each of its neighboring vertices v_j .



Figure 2.1: Compass direction drift over a large explored area. The ν estimate may deviate substantially (over 90 degree) from its starting value, but remains locally consistent.

Figure and caption reproduced with permission from Baumann (2019).

The overarching effort of this thesis is to reverse the accumulating angular errors ν_i from ν for each vertex of the graph of Baumann (2019) seen in figure 2.1. As the main requirement for our embeddings in all following sections we therefore establish

$$\forall i \in \{1, \dots, n\} : \nu_i = \nu \quad (2.5)$$

2.2 Assessment of embeddings

2.2.1 Problem

Given a graph $G = (V, E)$ with the adjacency matrix $A = \{a_{ij}\}$, where $a_{ij} = 1$ if $(v_i, v_j) \in E$ and $a_{ij} = 0$ else, and a set of connection angles ϕ_{ij} for all pairs $(v_i, v_j) \in E$ (or $a_{ij} \neq 0$), find an embedding $\mathbf{X} = (\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_n)^T \in \mathbb{R}^{n \times 2}$, $\mathbf{x}_i \in \mathbb{R}^2$ satisfying

$$\mathbf{x}_j - \mathbf{x}_i \approx \|\mathbf{x}_j - \mathbf{x}_i\| \cdot \begin{pmatrix} \cos \phi_{ij} \\ \sin \phi_{ij} \end{pmatrix} \quad (2.6)$$

$$\iff \vec{\mathbf{x}}_{ij} \approx \lambda_{ij} \cdot \hat{\mathbf{p}}_{ij} \quad (2.7)$$

$$\iff \hat{\mathbf{x}}_{ij} \approx \hat{\mathbf{p}}_{ij} \quad (2.8)$$

where

$$\begin{aligned} \vec{\mathbf{x}}_{ij} &:= \mathbf{x}_j - \mathbf{x}_i \\ \lambda_{ij} &:= \|\mathbf{x}_j - \mathbf{x}_i\| \\ \hat{\mathbf{x}}_{ij} &:= \frac{\vec{\mathbf{x}}_{ij}}{\lambda_{ij}} \end{aligned} \quad (2.9)$$

and the unit vector in direction ϕ_{ij} being denoted as

$$\hat{\mathbf{p}}_{ij} := \begin{pmatrix} \cos \phi_{ij} \\ \sin \phi_{ij} \end{pmatrix} \quad (2.10)$$

Notably, equation 2.6 only exists for such $i, j \in \{1, \dots, n\}$ for which $a_{ij} \neq 0$ and, therefore, ϕ_{ij} is given. Whenever $\hat{\mathbf{p}}_{ij}$ is used in the following, this exact notion would have to be implicit, which makes of poor mathematical syntax. Therefore, we establish

$$(v_i, v_j) \notin E \implies \phi_{ij} = 0 \quad (2.11)$$

2.2.2 Objective Functions

TODO: Mallot hat hier x_2 genommen, statt x_1 , um direkt anzudeuten, dass x_1 auf 0 liegt...

TODO: Das Quadrat wird im folgenden als die squared magnitude beschrieben, vielleicht sollte ich das nochmal mit Tristan und Mallot absprechen, ob das so gemeint war, außerdem vielleicht dafür ein Symbol einführen?

From the constraint equation 2.6 we can derive for a given graph G objective functions of the form

$$\Omega(G) : \mathbb{R}^{n \times 2} \longrightarrow \mathbb{R}, \mathbf{X} \longmapsto \omega \quad (2.12)$$

such as

$$\Omega_{init}(G, \mathbf{X}) = \sum_{ij} a_{ij} (\vec{\mathbf{x}}_{ij} - \hat{\mathbf{p}}_{ij})^2 \quad (2.13)$$

$$\Omega_{norm}(G, \mathbf{X}) = \sum_{ij} a_{ij} (\hat{\mathbf{x}}_{ij} - \hat{\mathbf{p}}_{ij})^2 \quad (2.14)$$

that assign a single numerical value, the *objective value* ω , to one specific graph embedding \mathbf{X} of a given graph G . Which objective value is implying a good embedding (see subsection 2.2.3) is dependent on the Ω it is calculated upon.

To get a straightforward assessment of the objective value in the normalized case of equation 2.14, we can fit this function to a range of values $W = [0, 1]$, basically resulting in an additional objective function. As $\hat{\mathbf{x}}_{ij}$ and $\hat{\mathbf{p}}_{ij}$ are both unit vectors with length one, equation 2.14 can be simplified through equation 2.15, which states as follows:

Be s, t unit vectors $\in \mathbb{R}^2$ and the scalar product given as $\langle s, t \rangle \in [-1, 1]$, then

$$\begin{aligned} \|s - t\|^2 &= (s_x - t_x)^2 + (s_y - t_y)^2 \\ &= s_x^2 - 2s_x t_x + t_x^2 + s_y^2 - 2s_y t_y + t_y^2 \\ &= \|s\|^2 + \|t\|^2 - 2 \cdot (s_x t_x + s_y t_y) \\ &= 2 \cdot (1 - \langle s, t \rangle) \end{aligned} \quad (2.15)$$

It follows that

$$\Omega_{norm}(G, \mathbf{X}) = \sum_{ij} a_{ij} \|\hat{\mathbf{x}}_{ij} - \hat{\mathbf{p}}_{ij}\|^2 \quad (2.16)$$

$$\iff \Omega_{norm}(G, \mathbf{X}) = \sum_{ij} 2a_{ij} (1 - \langle \hat{\mathbf{x}}_{ij}, \hat{\mathbf{p}}_{ij} \rangle) \in [0, 4\mathbf{m}] \quad (2.17)$$

$$\iff \frac{\Omega_{norm}(G, \mathbf{X})}{2} = \mathbf{m} - \sum_{ij} a_{ij} \langle \hat{\mathbf{x}}_{ij}, \hat{\mathbf{p}}_{ij} \rangle \in [0, 2\mathbf{m}] \quad (2.18)$$

$$\iff 1 - \frac{1}{2\mathbf{m}} \cdot \Omega_{norm}(G, \mathbf{X}) = \frac{\sum_{ij} a_{ij} \langle \hat{\mathbf{x}}_{ij}, \hat{\mathbf{p}}_{ij} \rangle}{\mathbf{m}} \in [-1, 1] \quad (2.19)$$

$$\iff \left| 1 - \frac{1}{2\mathbf{m}} \cdot \Omega_{norm}(G, \mathbf{X}) \right| = \left| \frac{\sum_{ij} a_{ij} \langle \hat{\mathbf{x}}_{ij}, \hat{\mathbf{p}}_{ij} \rangle}{\mathbf{m}} \right| \in [0, 1] \quad (2.20)$$

TODO: Can the norm of the last equation be taken so the range goes from [0,1] with 1 being good and 0 bad embeddings? 0 would be 90° across all scalar products, whereas 1 would mean either overlapping (0°) or mirrored (180°) bad embeddings. The latter might be swapped along the north direction to get the "more positive" partner embedding...?

which we will denote as

$$\Omega_{fitted}(G, \mathbf{X}) = \left| \frac{\sum_{ij} a_{ij} \langle \hat{\mathbf{x}}_{ij}, \hat{\mathbf{p}}_{ij} \rangle}{\mathbf{m}} \right|, W = [0, 1] \quad (2.21)$$

TODO: idea for a new approach for the objective function:
the length between each pair of vertices in relation to the length of the difference between the real and estimated endpoint given by the angular perception
 $\frac{\text{length of all? errors}}{\text{length of all? intervertex distances}}$

2.2.3 "Good" embeddings

As briefly stated in subsection 2.2.2, the meaning of a given objective value ω depends on the objective function Ω its calculated from. For example, an objective value of $\omega_{fitted} = 0$ would describe a *worst possible embedding* of perpendicularity between each pair of $(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{p}}_{ij})$ of a given graph G , whereas $\omega_{fitted} = 1$ would announce its perfect embedding. On the other hand, the perfect embedding for Ω_{norm} would be described by the value $\omega_{norm} = 0$. And for Ω_{init} there is no value for the *worst possible embedding*.

As can be seen, the notion of ω is not unified and has to be interpreted for each objective function differently. In addition, the construction of different objective functions can be founded on any measurable criterion of a graph embedding (not for the graph itself, as we look for the best embedding of one fixed graph). In our objective functions thus far we used as concept the *spatial*

distances between "should be" and "in fact" positions of the neighbouring vertices (see figure 2.2). Additional concepts could be *angular*, which evaluates the angular error rather than the distance, or *areal*, which spans areas by means of the vector(cross?) - product and thus combines the spatial and angular concepts.

2.3 Finding embeddings

2.3.1 Least square solution for Ω_{init}

For the non-normalized version of the objective function Ω_{init} in equation 2.13, the ideal solution should satisfy

$$\hat{\mathbf{p}}_{ij} = \vec{\mathbf{x}}_{ij} \text{ only when } a_{ij} = 1 \quad (2.22)$$

For the construction of the respective matrix form we will need the indexing functions

$$i(k) = \left\lceil \frac{k}{\mathbf{n} - 1} \right\rceil \quad (2.23)$$

$$j(k) = \begin{cases} \mod(k, \mathbf{n} - 1) & \text{for } \mod(k, \mathbf{n} - 1) < i(k) \\ 1 + \mod(k, \mathbf{n} - 1) & \text{for } \mod(k, \mathbf{n} - 1) \geq i(k) \end{cases} \quad (2.24)$$

With these we formulate equation 2.22 as

$$P = M\mathbf{X} \quad (2.25)$$

where $\forall k \in \{1, 2, \dots, \mathbf{n}(\mathbf{n} - 1)\}$:

$$a_{i(k), j(k)} \neq 0 \implies P_k = \hat{\mathbf{p}}_{i(k), j(k)}^T \quad \text{and} \quad M_{kl} = \begin{cases} -1 & \text{for } i(k) = l \\ 1 & \text{for } j(k) = l \\ 0 & \text{otherwise} \end{cases} \quad (2.26)$$

The dimensions of the three matrices are $P \in \mathbb{R}^{\mathbf{m} \times 2}$, $M \in \mathbb{R}^{\mathbf{m} \times \mathbf{n}}$, $\mathbf{X} \in \mathbb{R}^{\mathbf{n} \times 2}$.

With the construction of the matrices, we are able to compute directly the optimal embedding minimizing distances between $\hat{\mathbf{p}}_{ij}$ and $\vec{\mathbf{x}}_{ij}$ by means of the *method of least squares* (see ()). The *least square solution* is given as

$$\mathbf{X}^* = (M^T M)^{-1} P \quad (2.27)$$

For this method the $\mathbf{n} \times \mathbf{n}$ matrix $M^T M$ has to be invertible. If it is singular instead, we remove \mathbf{x}_1 from X as well as clear the matrix M of its first column, which will result in an implicit fixing of \mathbf{x}_1 to point $(0, 0)$.

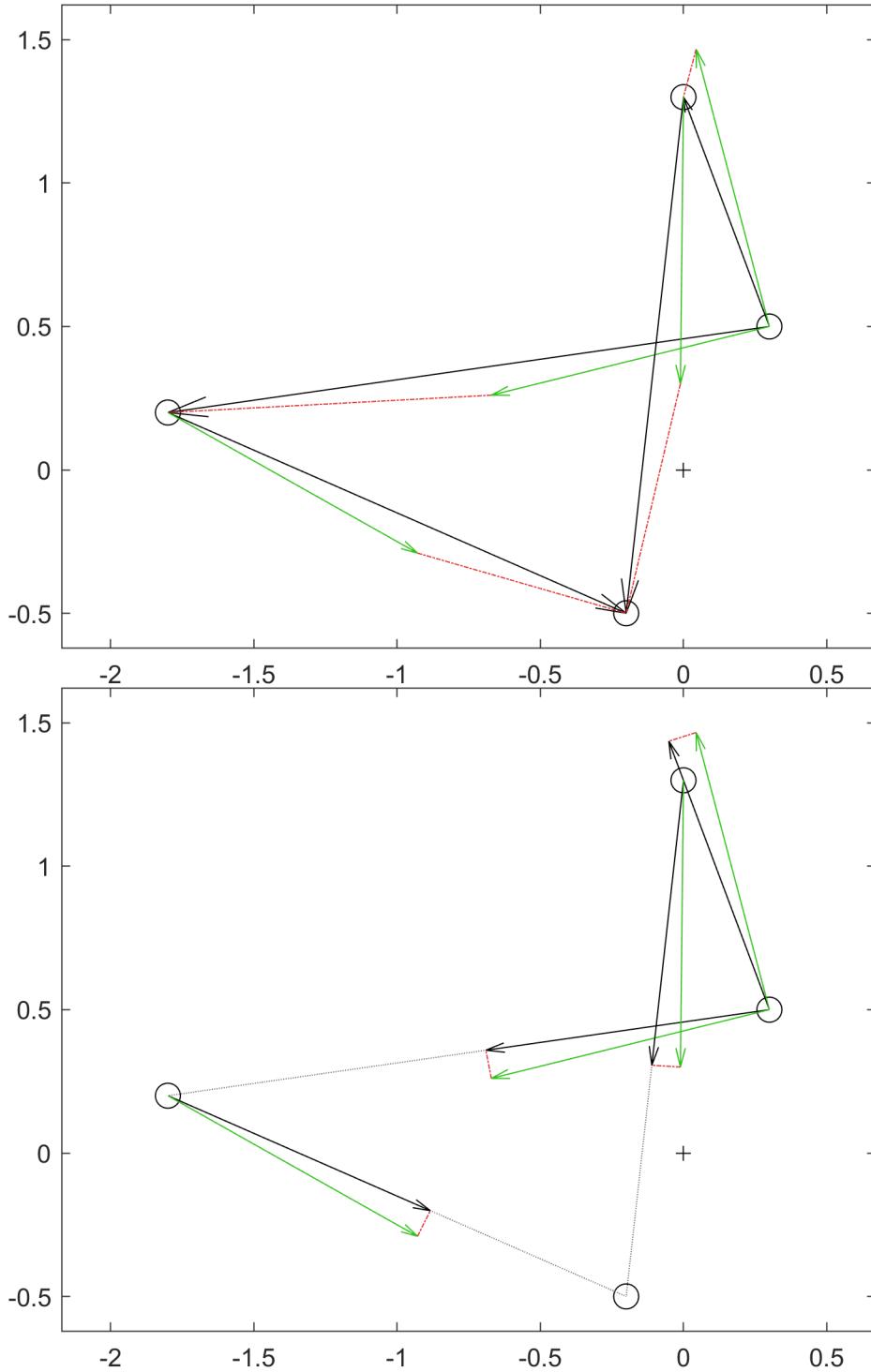


Figure 2.2: A random embedding of a directed graph with 4 vertices, its edges (black arrows) and the respective \hat{p}_{ij} unit vectors (green arrows) are drawn. These unit vectors are derived from the angular labels ϕ_{ij} of each vertex. **(top)** The differences between \hat{p}_{ij} and the non-normalized vector \vec{x}_{ij} are drawn as dotted red lines. This can be seen as the visualization of Ω_{init} (equation 2.13). **(bottom)** Here the edges are normalized and the differences drawn as dotted red lines are therefore of the respective unit vectors $\hat{x}_{ij} - \hat{p}_{ij}$ instead. Ω_{norm} (equation 2.14) is visualized here.

2.3.2 Gradient descent approach

Another solution for a good embedding could be approximated by means of an iterative approach called *gradient descent* (Baird & Moore, 1999; Lemaréchal, 2012; Ruder, 2016). There are many different versions of gradient descent algorithms. Dogo, Afolabi, Nwulu, Twala, and Aigbavboa (2018) or Ruder (2016) give some overview of commonly used gradient descent approaches.

In this thesis the gradient descent algorithm should update the embedding for the objective function to approach a local minimum (see again section 2.2.3). This minimal error is therefore dependent on the gradient of the loss function, in our case the used objective function.

While gradient descent and closely related topics such as *linear and multivariate regression* are very common for data analysis in various fields (e.g. machine learning, see Ogura (2017)), it comes with some limitations: As it depends on an initial configuration of the parameters (e.g. an initial embedding) the final solution might be a local rather than a global minimum of the loss function. In addition, a *learning rate* has to be chosen beforehand which greatly influences the convergence pattern, if the gradient descent converges at all. Both variables (learning rate, initial parameters) greatly influence the performance of the algorithm and need to be chosen accordingly for the gradient descent process to be of any use.

How it works

This section largely adopts and adapts chapter 1.5 in Nielsen (2015).

TODO: is this ok??

The main learning rule for gradient descent in our case will be

$$\Delta\Omega \approx \sum_i \frac{\partial\Omega}{\partial\mathbf{x}_i} \Delta\mathbf{x}_i \quad (2.28)$$

$$\iff \Delta\Omega \approx \nabla\Omega \cdot \Delta\mathbf{X} \quad (2.29)$$

where $\Delta\Omega$ is the change of the objective function when we make some adjustments to the input (the embedding). $\nabla\Omega$ is the gradient of the objective function Ω and defined as the vector of partial derivatives of Ω with

$$\nabla\Omega \equiv \left(\frac{\partial\Omega}{\partial\mathbf{x}_1}, \frac{\partial\Omega}{\partial\mathbf{x}_2}, \dots, \frac{\partial\Omega}{\partial\mathbf{x}_n} \right)^T \quad (2.30)$$

In addition we define the changes to our embedding $\Delta\mathbf{X}$ as a vector of changes for each vertex position with

$$\Delta\mathbf{X} \equiv (\Delta\mathbf{x}_1, \Delta\mathbf{x}_2, \dots, \Delta\mathbf{x}_n)^T \quad (2.31)$$

We want the change of the objective function $\Delta\Omega$ to be negative so that using this rule repeatedly will lead to a decrease in the objective value and over some iterations to a (local) minimum (Nielsen, 2015).

Let us choose the change in the embedding as

$$\Delta\mathbf{X} = -\eta\nabla\Omega \quad (2.32)$$

where the *learning rate*, which gives a measure of step size toward the current local minimum, is denoted as η . From equation 2.28 and $a^2 \geq 0$ for $a \in \mathbb{R}$ it now follows that

$$\Delta\Omega \approx -\eta \cdot \|\nabla\Omega\|^2 \leq 0 \quad (2.33)$$

With $\Delta\mathbf{X}$ being chosen according to equation 2.32, we therefore make sure that the objective function Ω always decreases.

The final *update rule* for our process of gradient descent sums up in

$$\mathbf{X} \rightarrow \mathbf{X}' = \mathbf{X} - \eta\nabla\Omega \quad (2.34)$$

Further reading on this topic more in the light of neural networks and backpropagation might be (Zell, 1994).

Gradients of objective functions

Let us now derive our gradients $\nabla\Omega$ for each objective function from section 2.2.2 respectively. The stepwise calculations can be seen in appendix A.1.

$$\nabla\Omega_{init} = -2 \cdot \sum_{ij} (\mathbf{x}_j - \mathbf{x}_i - \hat{\mathbf{p}}_{ij}) \quad (2.35)$$

$$\nabla\Omega_{norm} = \quad (2.36)$$

$$\nabla\Omega_{fitted} = \quad (2.37)$$

Choosing initial parameters

As for all gradient based optimization algorithms, we need to choose our parameters with care and some thought. In our case the initial embedding \mathbf{X}_0 , on which the algorithm starts, and the learning rate η need to be set prior to running gradient descent.

Let us design \mathbf{X}_0 in a manner similar to a breadth-first search:
TODO: cite something...

Following previous notations above, we can just number our vertices and respective embedding positions \mathbf{x}_i by ascending i , whereas any other ordering

might be possible and preferable (see below). We initialize our first \mathbf{x}_i at position $(0, 0)$. For all neighboring vertices we then compute the positions as

$$\mathbf{x}_j = \begin{cases} \mathbf{x}_i + \hat{\mathbf{p}}_{ij} & \text{if } a_{ij} \neq 0 \\ \mathbf{x}_i + \hat{\mathbf{p}}_{ji} & \text{if } a_{ji} \neq 0 \text{ and } a_{ij} = 0 \end{cases} \quad (2.38)$$

which sets these vertices on positions with distance 1 and at the optimal angle ϕ_{ij} or ϕ_{ji} from \mathbf{x}_i . We then repeatedly place one after another all neighbors for already placed vertices according to equation 2.38. This step is repeated until all vertices have initial positions.

The decision to design the initial embedding \mathbf{X}_0 as stated is mainly based on the consideration that, as such, \mathbf{X}_0 does have a gradient upon which the algorithm certainly goes toward minimization. As all positions are initialized to fit one angle ϕ_{ij} , the gradient descent algorithm can minimize the error generated by every other angle not considered in the initial embedding due to the ordering previous to it.

One might therefore consider using *simulated annealing* (see ...) to find a good random ordering which minimizes the initial objective value $\omega_0 = \Omega(\mathbf{X}_0)$. This reflects the possibility and increases the probability to begin gradient descent around the *global* minimum of the objective function Ω rather than a local one.

2.4 Application for "Virtual Tuebingen"

2.5 Embedding with Normalization

2.6 Applying the Theory

2.7 Comparing different graphs

2.8 Comparing different optimization strategies

3 | Practice

To apply the theoretical manners of constructing north-directed view-graphs from the previous chapter 2 in practice, a small survey was conducted. It was designed to collect data of subjective estimates of the direction from one location to another in a familiar surrounding. From the so called angular direction estimates (*ADEs*) view-graphs were constructed. The survey will be described in 3.1 and its collected data visually presented and analyzed in ?? . The raw data can be found in appendix ??.

TODO: Hypothesis

3.1 Survey design

3.1.1 Participants

The survey was performed by 6 participants (3 female and 3 male) being of age 13,19,22,27,51 and 66. Familiarity with the surrounding the survey took place in and locations thereof was confirmed by each participant. No monetary compensation was received, but oral consent has been given prior to participating.

3.1.2 Survey environment

The actual distances between each pair of landmarks were measured with the distance-measurement-tool provided by Google Earth (Google, 2020) and one angle between the landmarks Loeschenmuehle and Unterkaierberg was calculated with help of the suitable measurement tool (SunEarthTools, 2009) as 177.8° .

The respective distance matrix representing the pairwise landmark distances, we will denote it by D , is given in the appendix in table ?? . The survey surroundings therefore were confined within a radius of 2.6km around Loeschenmuehle, Bavaria. The maximum and minimum distances between landmarks for which angular estimations were done are 3.5km and 736m respectively.

With D it was possible to construct a metric embedding of the surroundings of Loeschenmuehle by means of multi-dimensional-scaling (Pich, 2009; Runkler, 2015; Torgerson, 1952). The calculated angle then was used to rotate the generated graph to fit the real world equivalent in orientation. In addition, the starting point X_0 , e.g. Loeschenmuehle, is set as the origin of the graph representation. This final embedding of used landmarks as well as their position on a map of the respective surroundings is given in figure 3.1. In the following sections this constructed embedding will be referred to as *correct embedding*.

3.1.3 Measurement of an angular direction estimation

All *ADEs* were measured by the participant itself by means of a compass, namely the "Präzisions-Kompaß für Sport und Freizeit von Eschenbach Optik, Nürnberg, Germany". This compass was simply used as a protractor, where the angles measured are in respect to the north direction of the compass.

To put it simple, the participants did execute angular measurements by means of a protractor (see figure 3.2):

The participants were positioned at one spot at the current landmark and were allowed to rotate around their own axis. Once in position, they were given the name of the landmark for which the *ADE* was to be measured. The participants then did rotate themselves to align their bodily *sagittal axis* (e.g. their egocentric view) along the estimated direction of the called landmark, while holding the compass in front of them. After adjusting the compass base to fit the compass needle, the *ADE* could be read off the rotary dial.

3.1.4 Procedure

The survey was constructed as a measure-travel-procedure (*MTP*) in the sense, that the participants conduct the desired measurements at one location and then travel to the next location to repeat these steps.

The exact procedure is explained in the following with 4 phases, from which phases 2 to 4 are repeated the necessary number of times. In this survey only 3 repetitions were done, as thereby collected data was sufficient for constructing view-graphs.

All angular measurements in the following are done according to 3.1.3.

Phase 1: Preparation

Starting at point X_0 , the participants were given the true north direction $\nu = 0$ as well as a list of n locations in the surrounding area by name. These locations

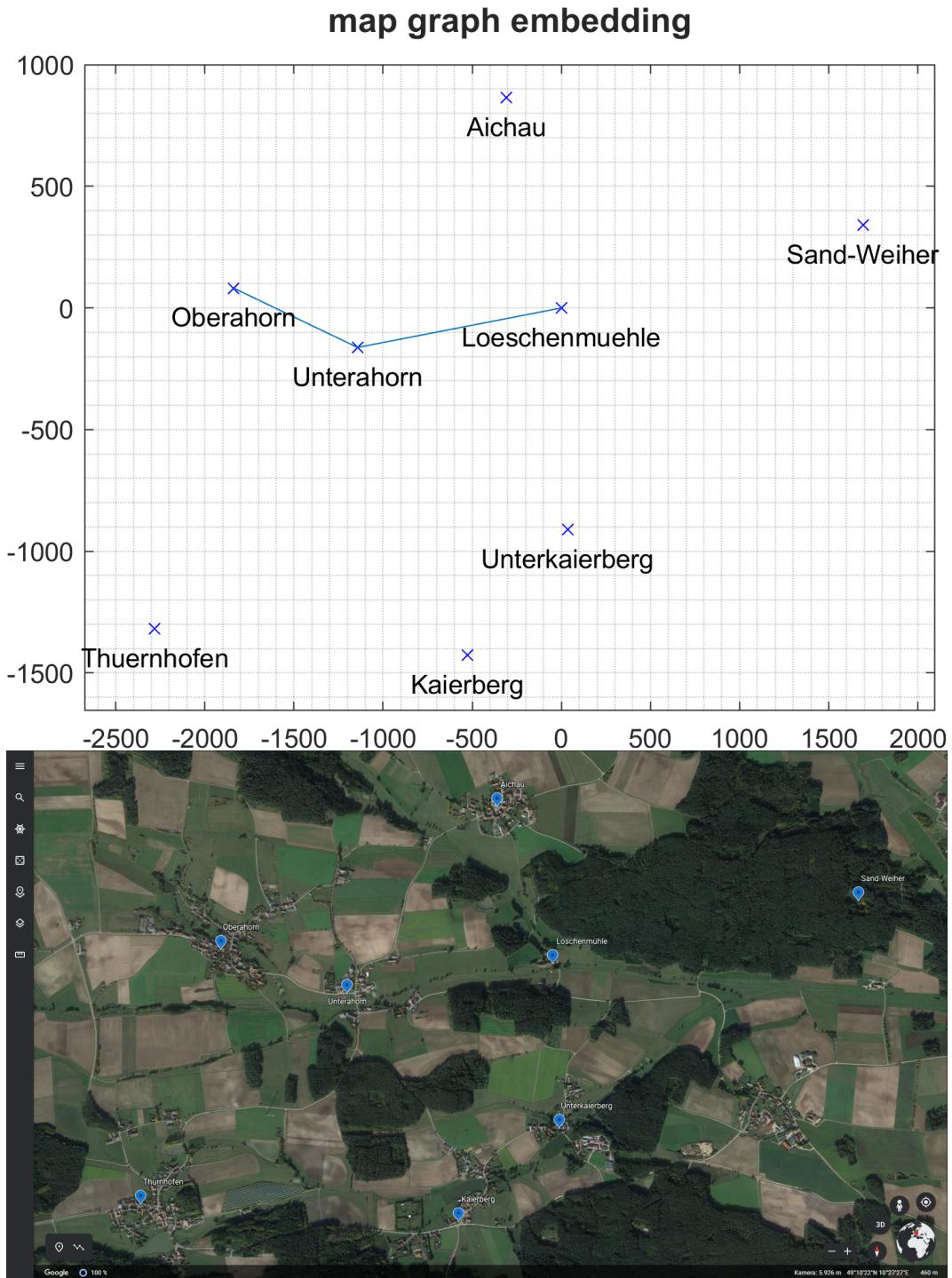


Figure 3.1: **top:** Graph embedding generated with distance matrix and angle between points 'Loeschenmuehle' and 'Unterkaierberg'. The route the participants did take is drawn as a blue line. Starting point was 'Loeschenmuehle' and endpoint 'Oberahorn'. **bottom:** Satellite map taken with Google Earth (Google, 2020), within which the respective landmarks are marked as blue flags.



Figure 3.2: Classic measurement of an angle by means of a protractor: **top:** Alignment of sagittal axis and compass along the angular direction estimation. **bottom:** Fitting the compass base to align the compass needle. The estimated angle can then be read off the dial.

along with the starting point are called *landmarks* and denoted by X .

$$X = \{X_i\} \text{ for } i \in \{0, \dots, n\} \quad (3.1)$$

Phase 2: North Estimation

Upon arriving at the current landmark X_i , the participants were to give an estimate ϕ_i of the true (compass) north direction ν . ϕ_i was first signaled through an outstretched arm, bodily alignment or fixation of an object in the visible surroundings and therewith compared to the true north.

Phase 3: Measurement

In the measurement phase the participants were to give an estimate of the angular direction for each landmark other than the one they were measuring at.

$$\alpha_i = (\alpha_{ij})^T \text{ with } i \in \{1, \dots, n\}, j \in \{0, \dots, n\} \text{ and } i \neq j \quad (3.2)$$

The *ADEs* for the landmarks were done in the same order over all participants.

Phase 4: Travel

The participants did travel between the 3 landmarks, at which measurement was conducted, either by car or bike. The routes taken between the landmarks were consistent over all participants regardless of the means of transportation. While neither blindfolded nor naive to their task, the participants were able to update their assessment of their surroundings and relations between landmarks during the travel.

At the locations of measurement (Loeschenmuehle, Unterahorn, Oberahorn) other landmarks (e.g. streets, tree lines, houses) were visible and internal odometric calculation (or its estimation) is to be assumed. One participant visibly imagined following the street toward the target landmarks, while others might have done so as well, although not in a visible manner.

The target landmarks were *not* visible at the 3 measurement locations.

3.2 Data visualization

In the following, in contrast to the correct embedding (section 3.1.2 or figure 3.1), we refer to an embedding we calculated from the collected angular data for any one participant (see appendix ??) by means of the least square solution 2.27 as a *subjective embedding*.

Figure 3.3 marks the differences between the correct and subjective embeddings. The subjective embeddings themselves can be seen in figure A.3 in

the appendix. Here we scale the correct embedding to equalize the distances between the points representing the landmarks "Loeschenmuehle" and "Unter-ahorn" for both embeddings. We rotated neither the correct nor the subjective embeddings and the vertices representing the landmark "Loeschenmuhle" in both embeddings are still at the origin point. This is done to be able to make an intuitive visual comparison. Below in section 3.3 this comparison is quantified.

The objective values (OV and nOV) are calculated with equations 2.13 and 2.21 respectively with the input given as the *ADEs* from each participant (see appendix A.2).

3.3 Data comparison

3.4 Conclusion

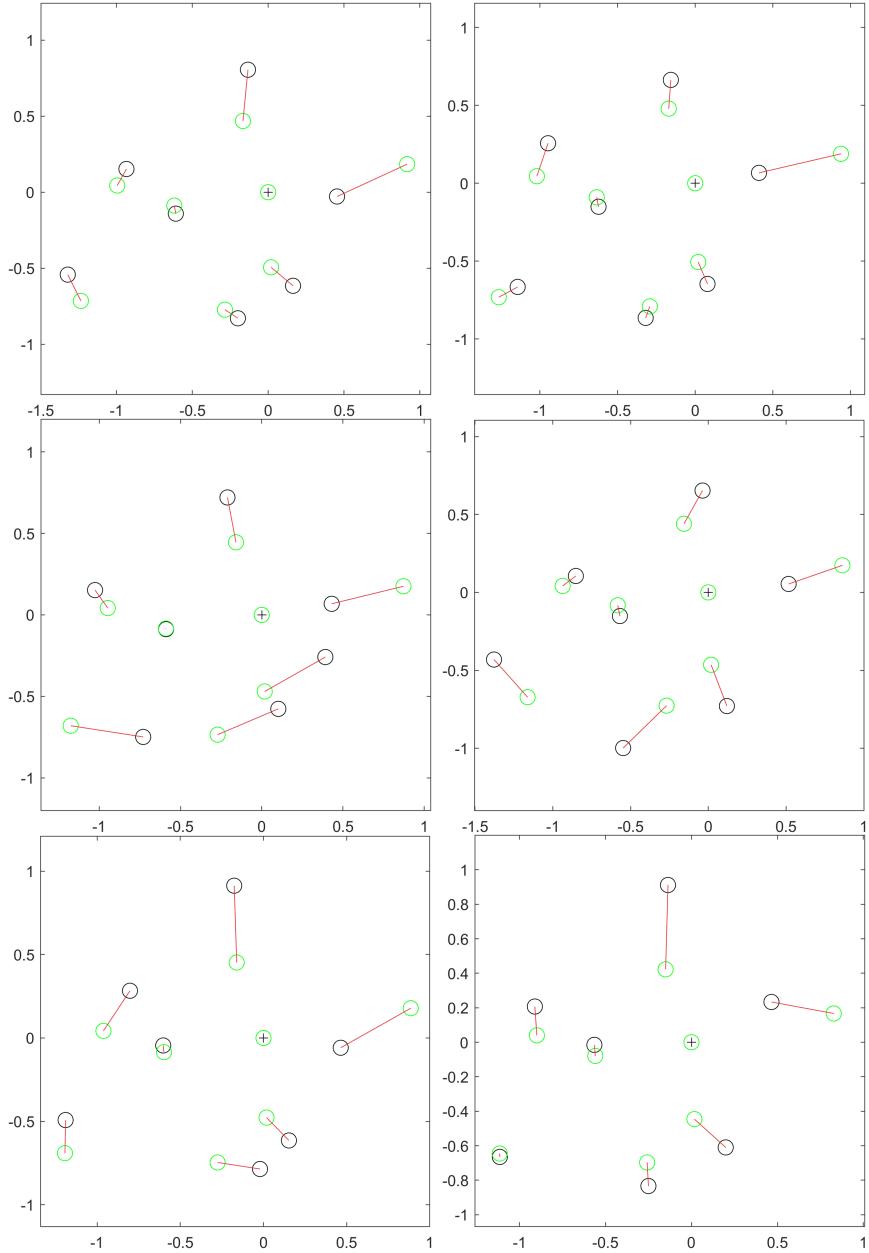


Figure 3.3: For each participant we compare their subjective embedding (**black** circles, see figure A.3) with the correct embedding (**green** circles, see figure 3.1). This correct embedding is scaled to equal out over both embeddings the distances between the vertices representing the landmarks "Loeschenmuehle" and "Unterahorn" (again see figure 3.1). No rotations have been applied and the embeddings are fixed at the origin. The **red** lines visualize the error between both embeddings for the same landmark.

4 | Discussion and Outlook

In future works on this topic we might explore this or similar procedures of finding good embeddings for other or specified graphs. We could assume weighted graphs (maybe representing multi-edge graphs as well) or specific graphs of the sort of planar, oriented, regular, complete, bipartite or more advanced cases.

References

- Baird, L., & Moore, A. W. (1999). Gradient descent for general reinforcement learning. *Advances in neural information processing systems*, 968–974.
- Baumann, T. (2019). *Dual population coding for path planning in graphs with overlapping place representations* (Unpublished master's thesis). University of Tuebingen.
- Bender, E. A., & Williamson, S. G. (2010). *Lists, decisions and graphs*. S. Gill Williamson.
- Diestel, R. (2017). *Graph theory*. Springer-Verlag Berlin Heidelberg.
- Dogo, E., Afolabi, O., Nwulu, N., Twala, B., & Aigbavboa, C. (2018). A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In *2018 international conference on computational techniques, electronics and mechanical systems (ctems)* (pp. 92–99).
- Google. (2020, Juli). *Google earth*. Retrieved from <http://earth.google.de/> (Accessed: 2021-03-23)
- Ishikawa, T., & Montello, D. R. (2006). Spatial knowledge acquisition from direct experience in the environment: Individual differences in the development of metric knowledge and the integration of separately learned places. *Cognitive psychology*, 52(2), 93–129.
- Lemaréchal, C. (2012). Cauchy and the gradient method. *Doc Math Extra*, 251(254), 10.
- Meilinger, T., Frankenstein, J., Simon, N., Bülthoff, H. H., & Bresciani, J.-P. (2016). Not all memories are the same: Situational context influences spatial recall within one's city of residency. *Psychonomic bulletin & review*, 23(1), 246–252.
- Moar, I., & Bower, G. H. (1983). Inconsistency in spatial knowledge. *Memory & Cognition*, 11(2), 107–113.
- Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 25). Determination press San Francisco, CA.
- Ogura, M. (2017). *Intuition (and maths!) behind multivariate gradient descent*. Retrieved from <https://towardsdatascience.com/machine-learning-bit-by-bit-multivariate-gradient-descent-e198fdd0df85> (Accessed: 2021-

- 04-04)
- Pich, C. (2009). *Applications of multidimensional scaling to graph drawing* (dissertation). University of Konstanz.
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.
- Runkler, T. A. (2015, 7). Data mining. In (2nd ed., p. 43-47). Vieweg: Springer.
- Stevens, A., & Coupe, P. (1978). Distortions in judged spatial relations. *Cognitive psychology*, 10(4), 422–437.
- SunEarthTools. (2009). *Tools for consumers and designers of solar*. Retrieved from <https://www.sunearthtools.com/de/tools/distance.php> (Accessed: 2021-03-23)
- Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4), 401–419.
- Yoder, R. M., & Taube, J. S. (2014). The vestibular contribution to the head direction signal and navigation. *Frontiers in integrative neuroscience*, 8, 32.
- Zell, A. (1994). *Simulation neuronaler netze* (Vol. 1) (No. 5.3). Addison-Wesley Bonn.

A | Supplementary data

A.1 Gradient computations

$$\begin{aligned}\Omega_{init} \\ \Omega_{norm} \\ \Omega_{fitted}\end{aligned}$$

A.2 Survey measurements

In the figures A.1 and A.2 all the numerical data for the survey conducted around Loeschenmuehle is listed. Namely these include the distance matrix D of the pairwise surrounding landmarks distances (in meters, A.1) and the estimates of angular directions from the 3 measurement landmarks (Loeschenmuehle, Unterahorn, Oberahorn) to every other (in degree, A.2).

A.3 Additional survey graphs

In the figures A.3 to A.6 different graph visualizations of the data in table A.2 are given. Graph embeddings in A.3 and A.6 are calculated analogous to the process described in 2.3.1.

	Sand-Weiher	Unterkaiерberg	Kaierberg	Thürnhofen	Oberahorn	Unterahorn	Aichau
Löschenmühle	1725.92	912.4	1519.91	2634.79	1840.3	1154.89	917.42
Sand-Weiher		2077.01	2836.04	4305.45	3539.84	2879.6	2066.88
Unterkaiерberg			761.46	2351.29	2120.43	1396.49	1808.77
Kaierberg				1757.09	1997.92	1405.71	2300.56
Thürnhofen					1468.92	1622.28	2943.1
Oberahorn						736.54	1719.53
Unterahorn							1324.01

Figure A.1: Table of pairwise distances between landmarks in the surroundings of Loeschenmühle. Distances are given in meters.

	Norden	Löschenmühle	Sand-Weiher	Unterkaienberg	Kaierberg	Thürnhofen	Oberahorn	Unterahorn	Aichau
VP1	0	-	84	164	198	254	270	264	346
	330	72 110	80 112	108 126	136 146	230 224	286 -	-	40 48
VP2	0	-	80	170	192	244	270	270	330
	350 18	70 118	68 117	110 136	137 172	226 194	278 -	-	39 76
VP3	0	-	81	117	175	233	267	288	327
	334 357	78 110	74 107	87 116	112 115	203 142	278 -	-	42 59
VP4	0	-	82	162	197	248	256	260	341
	0 90	82 102	80 96	110 148	174 182	238 250	279 -	-	44 70
VP5	0	-	88	162	170	264	278	268	340
	342 42	76 130	82 126	110 142	138 150	210 222	276 -	-	46 30
VP6	0	-	77	154	196	244	274	282	350
	356 4	78 116	65 98	117 132	150 150	208 210	277 -	-	33 44

Figure A.2: Collected data of angular perception for each participant (VP) at the 3 landmarks Löschenmühle, Unterahorn and Oberahorn in row-wise fashion respectively. All angles are given in degree ($^{\circ}$) from the compass north, e.g. in clockwise fashion.

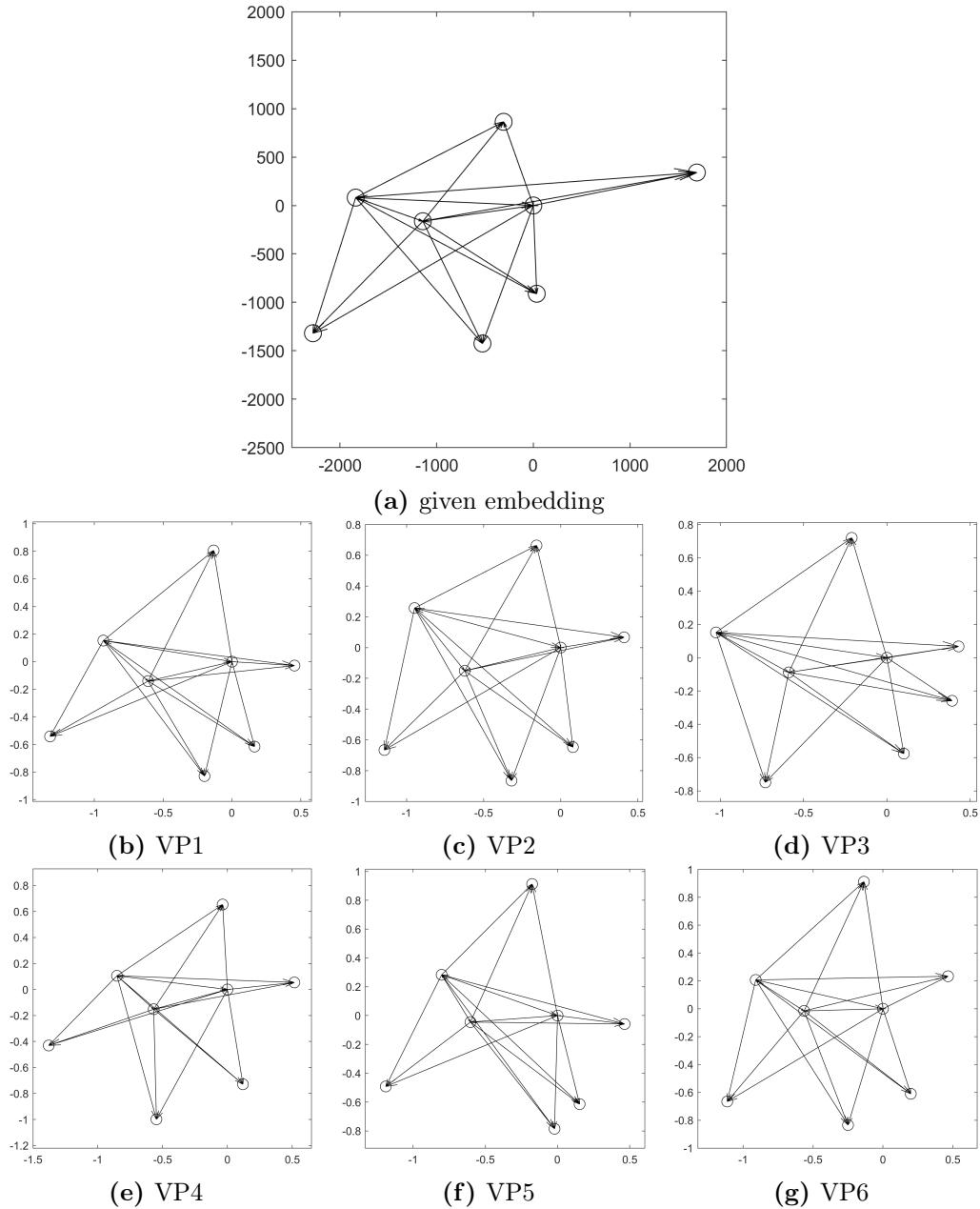


Figure A.3: (a) Graph visualization of the given map (see section 3.1.2 or figure 3.1) and (b)-(g) graph visualizations for all participants (VP) depicting the positions of the landmarks of the calculated embedding (black circles) and the edges between these landmarks (black lines). Note that the axes vary slightly here. Axes of (a) differ by a large factor. Calculation of embeddings is analogous to procedure of least square given in section 2.3.1.

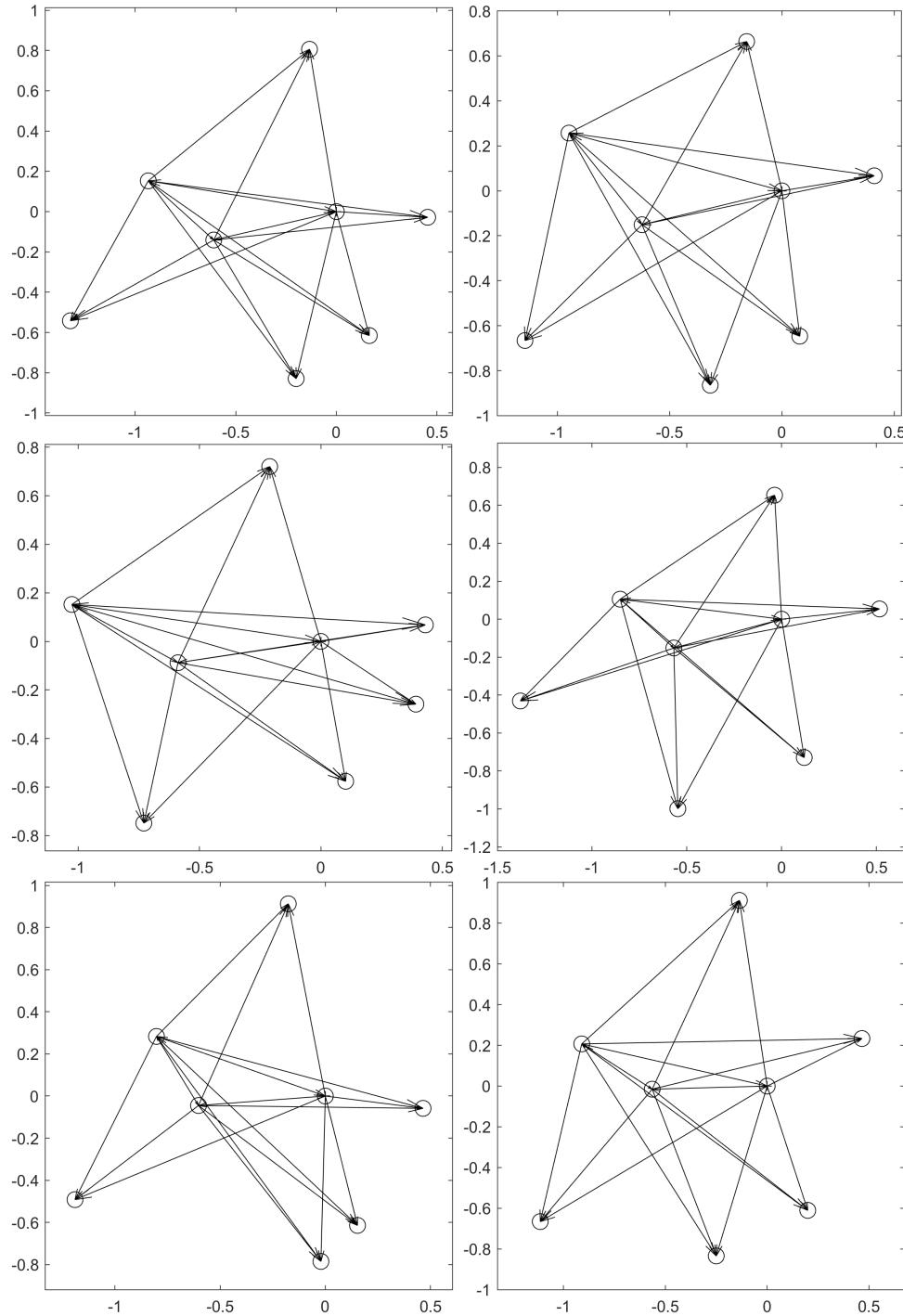


Figure A.4: Depicted are the subjective embeddings calculated by means of the least-square-solution (equation 2.27) and therefore minimizing the objective function 2.13. Each embedding is calculated from the collected *ADEs* of the respective participant. Note that the scaling of the axes do slightly differ.

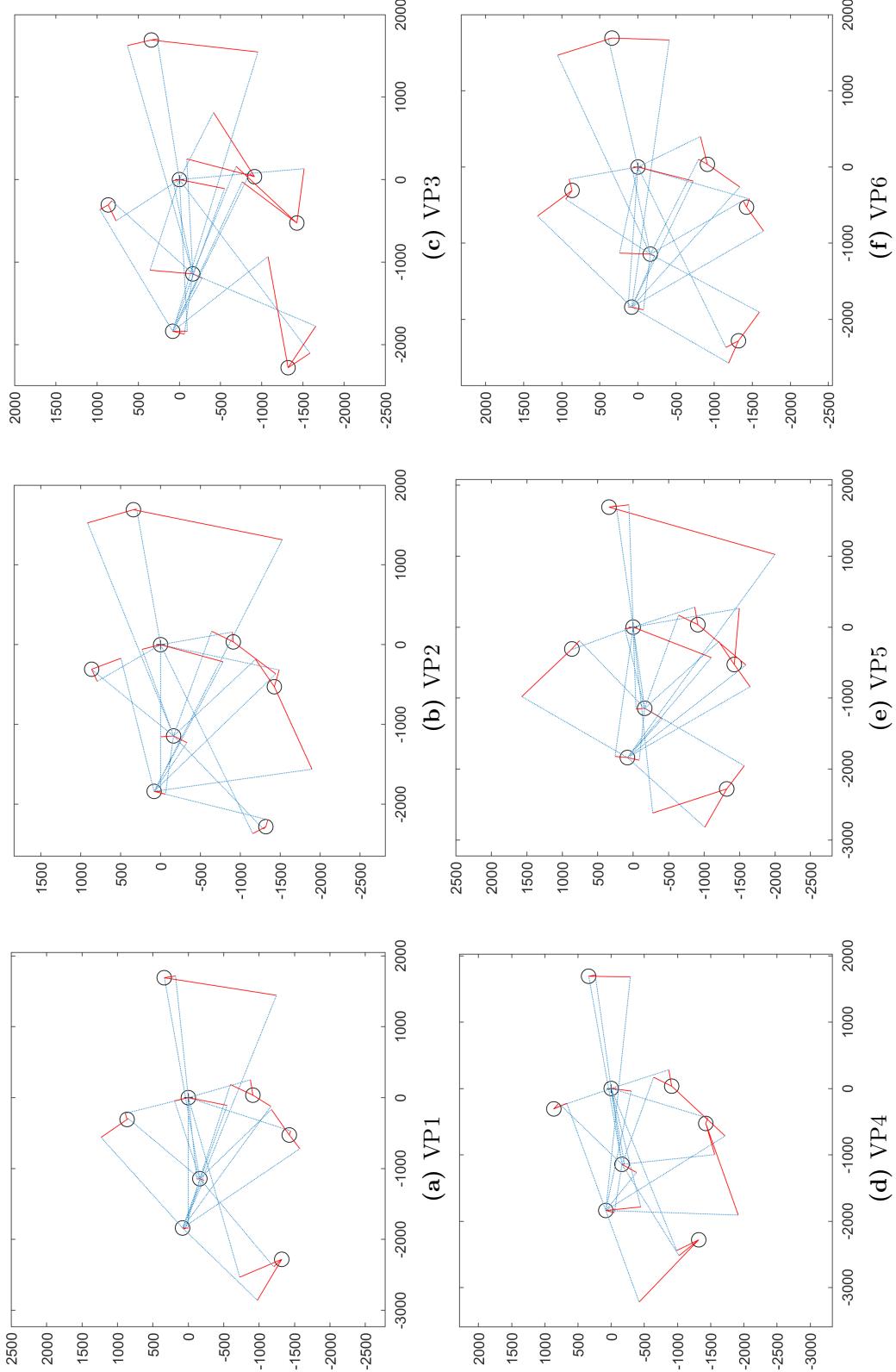


Figure A.5: (a)-(f) Graph visualizations for each participant (VP) depicting the positions of the landmarks of the correct embedding of the surroundings of Loeschenmuehl (black circles) and the angular direction estimates ($ADEs$) from the 3 measured landmarks as blue lines, whose length is equal to the distance to the target landmark. The red lines are therefore the error of estimated and targeted position.

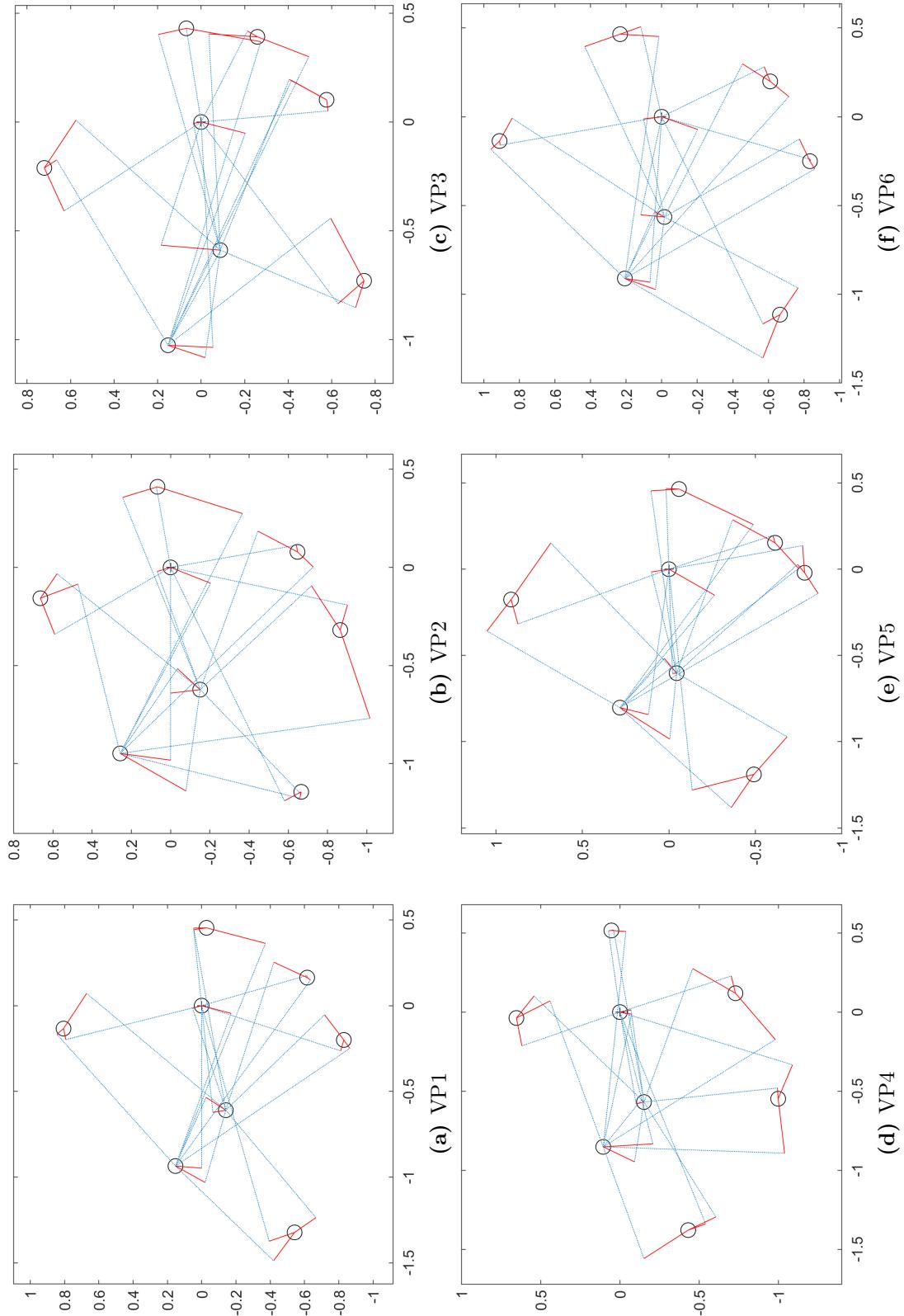


Figure A.6: (a)-(f) Graph visualizations analogous to figure A.5 with the difference that the underlying embeddings are the ones calculated from the data in table A.2 for each participant with the procedure of least square given in section 2.3.1.

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbstständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind.

Diese Bachelorarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Tübingen, den 11.4.2021

Jakob Meyer