Visualization of the Relevance: Using Physics Simulations for Encoding Context

Daniel Roßner*
Hof University
Institute of Information Systems
Hof, Germany
daniel.rossner@iisys.de

Claus Atzenbeck
Hof University
Institute of Information Systems
Hof, Germany
claus.atzenbeck@iisys.de

Tom Gross
University of Bamberg
Human-Computer Interaction Group
Bamberg, Germany
tom.gross@uni-bamberg.de

ABSTRACT

The task of organizing and retrieving knowledge is often elaborative and involves different types of media including digital or analog. In this paper we describe a system that is based on related research in the fields of spatial hypertext, information retrieval, and visualization. It utilizes a 2D space on which users can add, remove, or manipulate information entities (so-called *user nodes*) visually. A spatial parser recognizes the evolving structure and queries a knowledge base for helpful other information entities (so-called *suggestions nodes*).

Similar to user nodes, those suggestions are presented as visual objects in the space. We propose a physics model to simulate their behavior. Their characteristics encode the relevance of suggestions to user nodes and to each other. This enables human recipients to interpret the given visual clues and, thus, identify information of interest. The way users organize nodes spatially influences the parsed spatial structures, i.e., the placement of suggestion nodes. This allows the creation of complex queries without any prior knowledge, yet the users do not have to be aware of that, because they can express their thoughts implicitly by manipulating their nodes.

We discuss the strengths of a physics based simulation to encode context visually and point to open issues and potential solutions. On the basis of an implemented demonstrator we show the benefits compared to similar and related applications in the field of information visualization, especially when it comes to tasks where a high portion of creativity is involved and the information space is not well known.

CCS CONCEPTS

• **Information systems → Information retrieval query processing**; Multimedia databases; Search interfaces; • **Human-centered computing → Hypertext / hypermedia**; *Touch screens*; *Visual analytics*; *Information visualization*;

KEYWORDS

spatial hypertext; semantics visualization; Mother

*Corresponding author

HT '19, September 17–20, 2019, Hof, Germany

ACM ISBN 978-1-4503-6885-8/19/09.
https://doi.org/10.1145/3342220.3343659

ACM Reference Format:
Daniel Roßner, Claus Atzenbeck, and Tom Gross. 2019. Visualization of the Relevance: Using Physics Simulations for Encoding Context. In *30th ACM Conference on Hypertext & Social Media (HT '19), September 17–20, 2019, Hof, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3342220.3343659

1 INTRODUCTION

Spatial hypertext supports authoring of implicit structures by applying visual clues to express relation and is easy to read, as humans can take advantage of their perceptual abilities to navigate and explore space visually [21]. Effort went into the research of algorithms that make user-generated, implicit, spatial structure usable for hypertext systems. Starting with Marshall and Shipman, who implemented a parser, recognizing pre-existing structure types like piles, composites, alignments and hierarchies of those [19, 30]. Igarashi et al. introduced a genetic algorithm that adapts the parser, based on the users' feedback. Furthermore it improves interpretation of ambiguous structures like clusters [14]. With CAOS [25] it was shown, that it is advantageous to keep the machine's interpretation in sync with what the user is doing, because it allows instant visual feedback and further applications like collaboration support. Most recently Schedel enhanced spatial parsing with a temporal inspection of user interaction events [28].

The mentioned improvements show the difficulty of handling human-generated spatial structure. To come close to an interpretation of a human being, many properties have to be considered. It is not enough to just check for well-defined structure types or proximity. Moreover humans tend to interpret things differently, which means that different results may be equally good or bad.

In this paper we focus on the process which could be referred to as "inverse spatial parsing". Instead of interpreting spatial structure, the machine is able to generate it, in a way a human potentially would do it. While this is a very generic scenario, we narrow it down as follows: A system allows a user to add, remove and manipulate *nodes* in a 2D space. The "content" of these is not of interest for this system, they might represent software components in a development tool [27], query phrases of an information retrieval tool [15] or simple keywords. Comparable to VIKI [22], the system uses a spatial parser to interpret visual relationships between nodes and queries a knowledge base for *suggestions*. As with the nodes, the content of these depend on the domain the system is used for. They come together with a relevance measure from 0 to 1 in relation to each node, including other suggestions. The overall system is described in [4] and depicted in Fig. 1. The visualization of the suggestions happens within the 2D space that is used by the user, in such a way that it (1) does not alter the structure of user-generated nodes, (2) preserves the suggestions' structure and (3) merges both visual structures into one spatial hypertext, which is readable as another human would have added the suggestions. To combine the concept of understanding and producing emergent visual structure is a novel approach in spatial hypertext research, that offers an efficient platform of communication between human and machine. Users do not have to change their focus to, e.g., a list or something else; they can continue what they are doing and still profit from the system's input.

Whenever a scenario involves human creativity which can be supported by a machine, AI or a search engine it benefits from our proposed system. This is because of the often implicit and visually accessible nature of spatial hypertexts. Creative tasks consist of many manipulations and changes of ideas, documents, sketches and other types of "media". This media is organized in a space, may it be a hobby room, a sketch board or as in our proposal a computer screen. If the system is able to understand the organization of the bits and pieces of a creative task and integrates its help, suggestions and support in the same space, as an other human being would do, its output is easier to understand and probably of better quality.

The challenges of this visualization techniques are similar to that of spatial parsing: Many factors influence how humans interpret what they are seeing on their screen. Proximity, defined by the position of the nodes, is the most important one for any perceptual task [18], belong others like size, color and shape. Additionally the system should be able to react to any manipulation done by the user in real time to give immediate feedback in the form of an updated visualization and for significant structure changes updated suggestions. These updates should happen as often as possible, but it is important to offer a smooth, non-intrusive transition in between.

2 RELATED WORK

This work has evolved of research in spatial hypertext systems, spatial parsers, and antecedent systems became part of the hypertext community in the early 1990s. Systems like Aquanet [20], VIKI [22], VKB [29], or CAOS [25] propose structure mechanisms that are different to nodes and links, the most prominent structure type in hypertext, also known as navigational hypertext. In his 1987 paper, Frank Halasz criticizes the explicit nature of navigational hypertext systems: "The static nature of hypermedia networks could be largely eliminated (when appropriate) by including in the hypertext model a notion of *virtual or dynamically-determined structures*" [12]. Spatial hypertext is one possible way to follow this proposal. It provides such dynamic, virtual structures, as associations between nodes are represented implicitly. Similar to paper snippets on a desk, the spatial arrangements or visual appearance of nodes in spatial hypertexts reveal their associations only by interpretation. Such associations are fuzzy, fragile, and ambivalent by nature. As a result, different people may interpret those implicit associations or their respective strengths differently, which leads to challenges related to the automatic placement of nodes within a given spatial hypertext.

The visualization of retrieved suggestions is related to the extensive work done in the area of information visualization and retrieval in general. While some challenges are similar to that of graph visualization, e.g., nodes should not overlap, the solutions are not completely applicable. The most important difference is the lack of edges, because relations are "encoded" by visual means. Therefore algorithms do not have to care for potentially crossing edges. Furthermore they do not support that high degree of interaction needed and do not consider structure created by humans, which must not be altered, but augmented with additional nodes, only.

The field of recommender systems has produced great research on how to generate suggestions for product and services for users who do not need to be experts of the respective domain [26]. Here it has been shown that besides the algorithmic quality leading to adequate suggestions also models for presenting and explaining the recommendations are essential [5]. This holds for both single-user systems and group recommender systems. For instance, the RecoUIE system pro-actively generated recommendations and dynamically positioned them on the multi-user interface [11]. For positioning suggestions multiple visual attributes are potentially relevant (e.g., position, shape, or size) [6].

Olsen et al. introduced a system to visualize documents in a 2D space, called VIBE [24]. A user can define an information space consisting of a number of points-of-interest (POIs). These POIs have a similar meaning as our user-created nodes, they are used to retrieve documents, which are related to the POIs. Each document has a score/relevance to each POI, that is used to compute a position by calculating a linear combination. If a document depends on one POI only, it is placed right on top of it; if it is equally relevant to two POIs it is placed in the mid of the line between those. Moving the POIs around is a very important task for the user, because it may reveal related logical clusters of POIs and documents. Our system differs from that, as it does not only honor the position of "POIs" (nodes), but their overall visual structure and we do not represent "documents" (suggestions) as points of different sizes, which may overlap and can not be distinguished, but as objects that can contain any type of representation. Fig. 4 shows two screenshots, where the content is represented as text, picture and video. Moreover VIBE focuses on relations between documents and POIs, it omits those between documents. But this might be a relevant information when it comes to discovering an unknown information space, because it helps finding relevant clusters of information, even though they are barely related to the POIs. The concept got improved by Klouche et al. [15] with an additional result list view and the capability of visual re-ranking the result. Documents are rendered as opaque circles of varying radius. The opacity encodes the density of documents and radius the overall relevance. By clicking anywhere in the visual space, the result list gets re-ranked, such that "closer" documents are shown first. As VIBE this system does not make relevance of documents to each-other visible for the user. After all, it is a trade-off between losing structural information and ambiguity when too many relationships influence positioning.

Systems like Bead [7] utilize a 3D space to represent visual structured documents. In contrast to VIBE, Bead does not support "POIs" within the information space, but only as words which form a query. Document distances are measured, the more similar documents are, the smaller the distance is. Bead renders documents as simple particles and uses the measured distances to apply forces of attraction

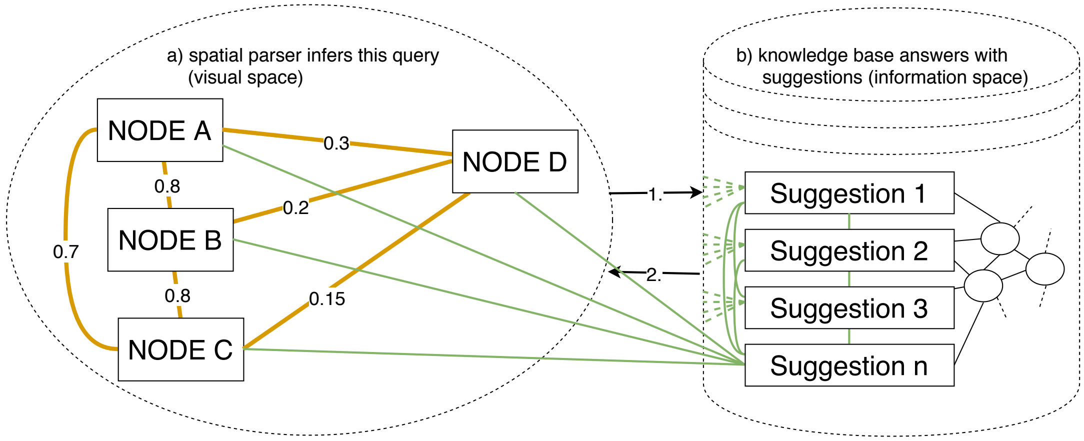


Figure 1: a) shows the 2D visual space with four nodes A, B, C, and D. Orange colored relationships are calculated by a spatial parser. The higher the value, the more visual related nodes are. The result is sent to a knowledge base, searching for related suggestions in the information space (green relationships, not all are shown here). Those green relationships have a weight between 0 and 1 to indicate relevance an as such build – with nodes A to D – a complete graph, which is sent back to the visual space. In a not shown step, suggestions are rendered within the visual space.

and repulsion to all the particles, based on the model of a damped spring. Following the minimum total potential energy principle the system of particles tends to a state of minimal potential energy, meaning that all springs are as close to their "preferred distance" as the system as a whole allows. The relaxed position of a spring is derived by document distances, meaning that a user can use his perception of proximity to recognize strongly related particles. Even if not utilized in this system, the physical based simulation allows real time calculation of positions, while a user is manipulating particles. We adapted that concept for our prototype and added awareness of pre-existing structure.

With InfoCrystal [31] Spoerri presented a solution for some problems of VIBE and BEAD. If one is using big queries, e.g., many POIs in VIBE, multidimensional relations have to be scaled down. InfoCrystal is inspired by Venn diagrams, but imposes a structure to overcome its limit to represent relationships among more than three sets. Instead of showing documents directly, they are hided behind an iconic representation, one for each combination of possible boolean queries. By (de-)selecting those inner icons, the output can be filtered accordingly. This solution simplifies rendering and querying of documents, at the cost of losing valuable information, like the continuous value of relationships to each other. Instead we propose to reduce the number of rendered suggestions initially by calculating the relevance to a complete set of nodes, e.g., the mean value, and taking the best n. After that, it is the user's responsibility to change n or mark suggestions that she is not interested in, to load more of them.

This type of interaction can be seen as a sort of visual querying. Users start with a simple statement and receive suggestions, whose relevance is visual encoded. If they are interested in one or more of them, they can add them to their own nodes, thus alter the scope of the original query. A similar system is used by VINETA [17], which limits the interacting to clicking on keywords. The more keywords are marked, the less the impact of antecedent keywords is, regarding the result. In our prototype this can be achieved by changing the visual structure, which gives the user more control on what is happening in the visual space.

DARE [34] provides the user with a special view, the so called "visual space" to display the multidimensional document space in 2D. Users build a query vector and choose a reference point, which can sit anywhere and is movable at any given time. The two orthogonal axis in the visual space represent a (1) distance and (2) cosine measure. The angle is formed by two lines, starting at the document vector and the freely chosen reference point, both intersecting at the query vector. The model is suitable for different evaluation models, which essentially shows the strength of a combination of different measures in one visualization environment. This concept is carried on by TOFIR [33], using an angle-angle based visual space, whereas GUIDO [23] promotes a distance-distance based one.

In contrast to prior work, we try to broaden the scope of the system by combining concepts from spatial hypertext research, especially visual/spatial parser to give the machine a sense of what a user is expressing spatially, and visualizing techniques of information retrieval tools. Apart from that, we demand interaction and manipulation of the information space to be a very important possibility to refine queries.

3 SEMANTICS VISUALIZATION

When we talk about semantics in context of hypertext, we usually mean relationships. Many characters can form words, words can form sentences, if they are seen in a certain order. Relationships reveal semantics, which is defined by a grammar in this example. Before describing our conception of semantics visualization, we need to specify the kind of hypertext we are dealing with. We assume a knowledge base of domain specific information. This information space is a graph, whose edges encode relevance between nodes – missing edges indicate no or zero relevance. We do not assume higher order structures like hierarchies and composites. Whenever

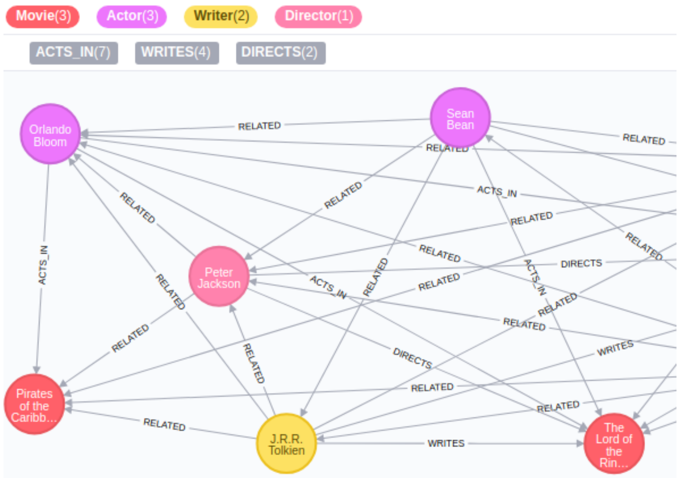


Figure 2: Excerpt of a knowledge base for demonstration purposes. Contains weighted relations of movies, actors and other participants.

such relations are needed, they have to be encoded elsewhere, e.g., "within" the node or an other structure service comes into place [3]. As in Fig. 1, relevance is represented as numerical value between 0 and 1, called *weight*. The calculation of those depends on the domain, e.g., for documents the vector space model could be applied. In Fig. 2 we show an excerpt of the knowledge base that is used for the demonstrator described in Sect. 4.

We aim to provide a system, that is able to visualize these relations. In Sect. 2 we describe several other solutions, that demonstrate the strength of visualizing such a structure in a 2D space [1] to facilitate human visual perception. Additionally we see an important role in representing the relation between a query and its results. It is a try to merge the visual space of the user with the information space of the machine. Both worlds are hard to understand for each other. Therefore we propose spatial parsers and semantic positioning as tools to allow both partners to speak the same visual language (cf. [13]). This additional communication channel is easy to use, as it does not demand for a special syntax, because it takes advantage of the implicitly happening interpretation of visual structures.

3.1 Physics metaphor

Initially we experimented with simple and naive positioning algorithms. A circular layouter arranges information objects radially around a geometrical center area, resembling a fish eye view. Very relevant suggestions are close to the center and bigger than less relevant ones. While this works quite well to identify suggestions with a high relevance to a query, which is located in the center area, the structure of those is not visible. In a next step, we adapted the idea of VIBE, by adding a factor, representing the degree of suggestions overlapping each other. An optimizing algorithm tries to find a good compromise of calculated position and overlapping factor. It turned out that this approach does not work well in an interactive environment: moving nodes may lead to a situation, where compromises are not good anymore. Any re-optimizing can result in re-assigning completely different positions. Suggestion start to jump around, making it impossible for the user to track their path. Furthermore, as in VIBE, the rendering area for suggestions is confined by the user's nodes. Another approach tackled this issue by splitting a suitable big area around objects forming the query in many smaller rectangular pieces. For each of those subareas a relevance value was calculated, meaning that suggestions of a certain relevance would fit in there. This value honored the distance and obstacles between a node of the query and the sub area. The greater the portion of obstacles to the total distance, the less the value. Heat emitting objects would be a good metaphor: Thermal radiation decreases over distance and is absorbed my massive objects. But the more sophisticated the positioning algorithms got, the harder it was to provide a decent behavior for interacting scenarios.

Many graph visualization tools solve a similar problem, by implementing a physics metaphor [16, 32]. Instead of putting much effort in complex algorithms, a simulated physical environment is created. It is not necessary to have a high accuracy or to support all different kinds of mechanics. Physical attributes, like density, velocity, friction, or size are assigned to any visual object, defining their behavior in the simulation. Repulsion can prevent overlapping, attraction keeps objects together and so on. There are many metaphors out there, which give a sense of how visualization helps us to achieve a diverse set of tasks. Physics metaphors make use of the "naturalness [...] based on the everyday familiarity of the physical environment" [8].

3.2 The applied model

When choosing a physics metaphor, it is important to apply a suitable model of physical attributes to all visual objects, such that it matches our definition of semantics visualization. As proximity is by far the most important and generally applicable attribute to encode relationships, we focus on that.

Following the example of Bead, we use a spring based model to attract and repulse objects to and from each other. Which kind of force is applied depends on the current length of a connecting spring. A spring has an ideal state, a relaxed length with no potential energy. Thus the corresponding relevance must be mapped to this state. In Fig. 3 all important variables for that mapping are illustrated. The perception of proximity always depends on the given spatial context (cf. [10]). The "bigger" two objects are, the higher the absolute distance can be and still be recognized as close to each other. Therefore a factor d_{FAC} is introduced as

$$d_{FAC} = -rel \times (d_{MAX} - d_{MIN}) + d_{MAX}$$

where d_{MAX} refers to a distance factor that indicates a relevance of 0 and d_{MIN} of 1. The result is multiplied by the sum of r_a and r_B, giving us an absolute value for the ideal length of the underlying spring model. Because this spring is connecting the centers of both objects, d_{AB} is calculated as follows:

$$d_{AB} = d_{FAC} \times (r_A + r_B) + \frac{r_A + r_B}{2}$$

Whenever a spring is not in its ideal state, it applies a force on both objects, which increases the higher the deviation is. Additionally we use a greater force for high relevance springs. This improves the positioning of important suggestions, but may lead to a worse positioning of irrelevant ones.

As there are nodes, whose position and movement is controlled by the user only, the spring model may calculate states with overlapping. To avoid such situations, all bodies react to collisions, such that they do not overlap and pass their impulse; user nodes do not react do any impulses and do not collide with any other object, allowing users to build piles and stacks of objects.

The spatial parser generates queries by identifying visual strongly related nodes and sending them to the knowledge base. Thus, suggestions are based on visual “clusters”. To improve positioning on a cluster basis, the spring model is applied to clusters only, meaning that suggestions which correspond to different clusters would not influence each other. It turned out that, as the user is not able to see explicit connections, this leads to misinterpretation of the underlying semantics. Regarding the physics metaphor we added a general force of repulsion to all suggestions in the visual space. On a cluster basis this force has a very small impact, because spring forces keep things together as intended, but as this is the only inter-cluster force, suggestions of a cluster tend do move away from other clusters. You can see this effect in Fig. 4a. where suggestions for “Brad Pitt” float to the right. Moreover we added color as additional visual clue.

Movement and interaction influence the environment and thus the physics in the simulation. How the system reacts, depends on the applied forces, explained above and (1) weight (defined by size and density), (2) friction to damp velocity and (3) the coefficient of restitution when objects collide. The values for these are slightly arbitrary, as they influence each other and depend on the scaling factor, which transfers meters to pixels, but we tried to achieve the following behavior:

(1) The system should be responsive to user interactions but be settled promptly when users stop interacting.
(2) Changes should be traceable, slow enough to follow, fast enough to keep up with the user.

A physics simulation is susceptible to changes that do not happen in the real physical world. Thus, the addition and removal of objects is a crucial point in our model. A removal is handled by deleting the object and all corresponding forces, including those applied by the spring model. An addition is more complicated, because the system needs to take care of the initial position of objects. If the initial position is far away from a desired position, the simulation will need some time to “move” the added and all other objects to an acceptable state. Furthermore this process looks agitated, not traceable for the user, especially when more than one object is added to the visual space at the same time. The first idea was to run a “hidden simulation” in the background. This eliminates the time delay, since its computation is not bound to real time. The downside is, that objects appear in the twinkling of an eye, without offering a feeling for evolvement. For that reason all objects added by the system (suggestions) are positioned in the geometric center of the visual cluster. The resulting simulation looks like an explosion, pushing the objects to their desired position.

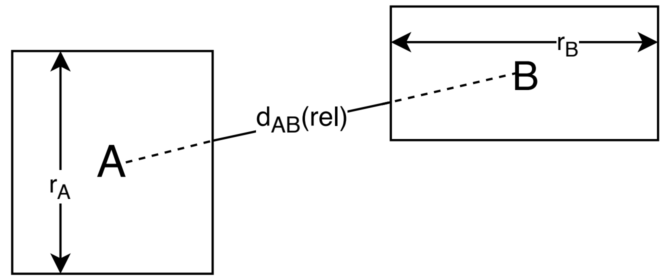


Figure 3: The ideal distance d_{AB} between A and B is calculated with the help of the height or length of a body, whichever is bigger (r_A and r_B). The relevance *rel* is converted to a factor

4 DEMONSTRATOR

4.1 Application domain

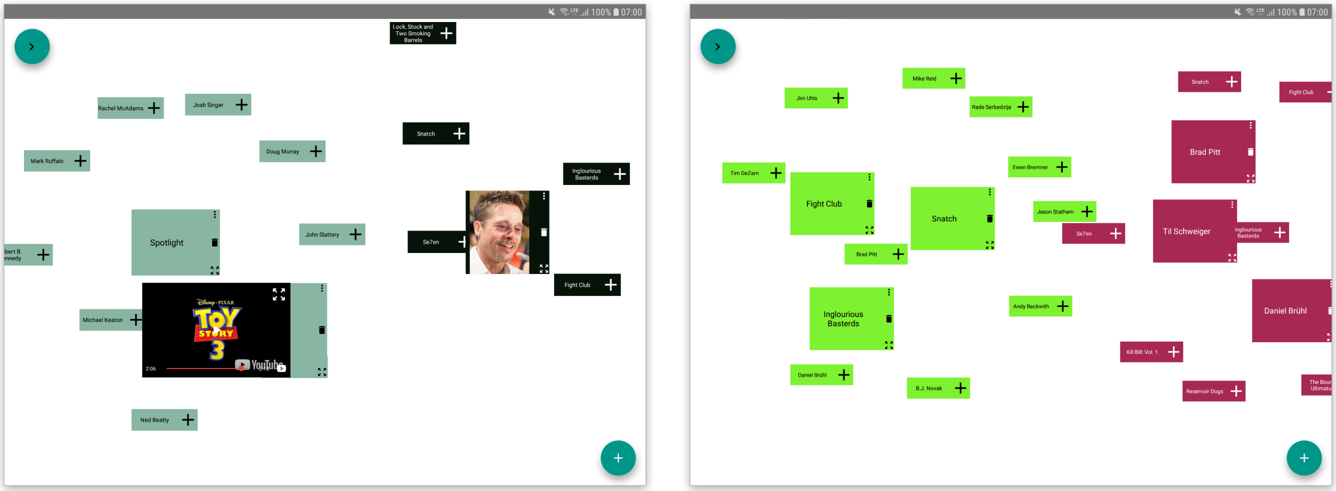
The *DemoMedia* project – developed in cooperation with *Loewe Technologies GmbH*, a manufacturer and seller of consumer electronic products – is an implementation of the proposed system in the field of entertainment. Its goal is to offer an application users can use to organize and search movies, actors and other things related to films and their production. A 2D space serves as room for organization and is augmented with suggestions of the system. Organization means, that a user can add, move, resize and delete information entities. In this demonstrator the user types some text to add such an entity. The entity is represented by a rectangle in the visual space. If the system can find further information to a typed keyword, e.g., the name of an actor, the user can, instead of just showing the text, change the type of representation to any media the application is offering. Currently we support pictures and content from *YouTube* (see Fig. 4a). Suggestions are visual rectangular objects, which contain information about movies and actors. The user is not explicitly asking for these suggestions, they are generated implicitly by a spatial parser, as described in Sect. 3.2. Suggestions and user nodes can be distinguished by their size and (missing) control bar. User nodes offer an option to resize, delete and open other settings on their right, while suggestions are smaller by default and offer an “+” control to “convert” a suggestion to a user node.

The underlying data and knowledge base is crawled and computed from

(1) *IMDb* (Internet Movie Database),
(2) television program dump provided by *Gracenote* and
(3) *YouTube*.

4.2 Use case

Users are searching for films they do not know yet, but might be interested in. They start, like in common systems, with a keyword. This keyword stands for a genre, year, an actor they like or anything that helps finding related results. Those are shown visually within the 2D space, such that the relevance to that keyword and the relevance to each other is modeled by proximity and the physics based behavior when interacting. This helps the user to oversee



(a) Shows three nodes: Spotlight, Toy Story 3 and Brad Pitt. The first two are recognized as visual related by the spatial parser, which sets the same color for them.

(b) The spatial parser identifies two visual related structures with three nodes each.

Figure 4: DemoMedia screenshots

the structure of suggestions. They can refine their query by adding more nodes with keywords, either by typing them on their own, or by pressing the "+" button of a suggestion and dragging it to any position they want it to be, like shown in Fig 5.

The process evolves in an iterative dialogue between user and the system, where structures emerge, get altered or destroyed. Each manipulation changes the positions of suggestions or introduces new ones. Instead of using complex query languages like *SQL* or filters that are hidden in drop-down menus and sliders, the user does not have to think about how to narrow his query down. Since the space can be panned and zoomed, the user is able to develop many queries in visual unrelated areas, which can be merged when he chooses to bring them together. Additionally the system offers controls to specify the number of shown suggestions and filter for movies or actors.

If the user is interested in details, the representation inside the nodes can be changed. This can be a wiki page, a slide show or the movie itself, which can be cast to a TV. It is not necessary to leave the workspace to interact with this content, as it is directly embedded. To improve interaction the object can be resized, zoomed, or brought to full screen. Thus, searching and consuming do not interrupt each other. A user can, e.g., listen to the theme music of his favorite movie, while reorganizing nodes in the workspace, or reading some critics. In a next step it is conceivable that this multi modal information is added to the query, generated by a multi modal parser, which should lead to more relevant suggestions.

The workspace is not reset when starting a new session, therefore the user is not forced to rethink and rebuild his associations. This fosters the motivation to structure information visually. The system is also able to profit of this evolving structure, because it can improve its stored relevance values, based on parsed visual relationships. Other users might take advantage of them afterwards.

4.3 Implementation

The developed demonstrator is a part of a larger set of services we developed over the last few years, called *Mother*. Simply explained, we divide the whole application in three pieces, (1) *Hel*, the knowledge world, (2) *Asgard*, the world where spatial parsing happens and (3) *Midgard*, place for user-centered applications that utilize Asgard and Hel components (cf. [4]). Hence, the demonstrator is a Midgard application, handling user-interaction, the user-interface and communication with server-based Asgard/Hel layers. Overall, Mother is a component-based open hypermedia system (CB-OHS) with well-defined communication protocols. This increases the system's flexibility to include new services or user interfaces.

The demonstrator is aiming to reach people, that make use of it at their homes sitting in an armchair. No technological affinity should be needed to get started. We chose to implement the client for tablet computers, because they offer a screen that is big enough to profit from the proposed representation and a touchscreen for a natural interaction with objects that have a proper physical behavior. To reduce the need to type text with the on-screen keyboard, the demonstrator offers text completion, thus the user just has to type the first three letters. Beside this, many devices and operating systems allow to input handwritten text (with a pen) that gets converted.

Whenever the user manipulates anything in the visual space, the application is sending a binary encoded message via TCP/IP to Asgard, that handles spatial parsing and query generation. The query is sent to the knowledge base, which is implemented by a low overhead component, proxying a *Neo4j* graph database. When

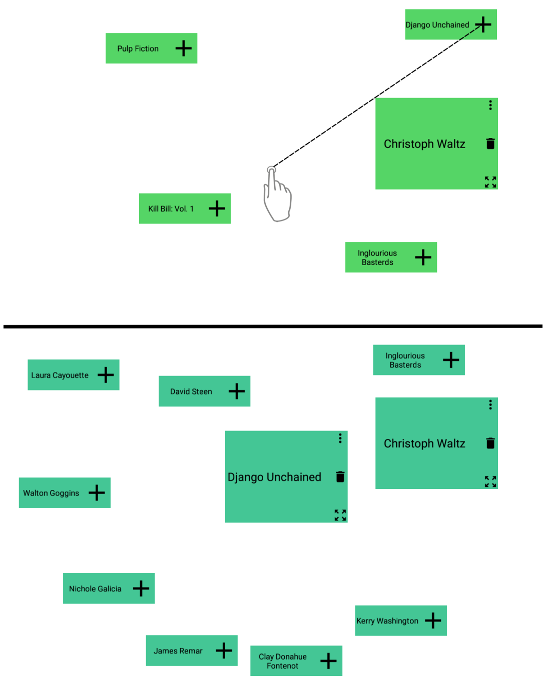


Figure 5: Suggestion "Django Unchained" is dragged to a user defined position. As it is close to "Christoph Waltz", the spatial parser interprets both as visual related, thus they become part of the same query.

the query is processed, the answer is sent back to the application, which takes care of the representation and physics simulation. For dragging events the application is not just propagating the final position of a object, but many intermediate steps, as they might change the parser's interpretation. The frequency of updates sent depends on the network's speed and delay, the spatial parser and knowledge base. Since the application requires responsive server components that are able to communicate without any disruptive lags, it optimizes communication by caching and adapting the rate of updates per second (sent to the server) to the average round trip time of an event and its corresponding answer. The update rate is capped to 60 events per second, which is hold most of the time using a WiFi-connection.

The physics simulation is done by *JBox2D*[1], an engine for simulating rigid bodies in two dimensions. The use of such a fully fledged engine allowed us to focus on experimenting with physical properties and forces. As an engine that was developed for games, it provides a good trade-off between speed and accuracy. With JBox2D in hand, it was easy to apply the proposed physics model of Sect. 3.2. The spring model is covered by the usage of so called *distance joints*, which forces two objects to a specified constant distance. Those joints can made soft by assigning them with a frequency and damping value. The higher the frequency value is set, the more flexible the imagined spring is and vice-versa.

[1] "JBox2D is a close Java port of Erin Catto's excellent C++ Box2D physics engine and Google's LiquidFun physics Engine", http://www.jbox2d.org/

5 EVALUATION

The following evaluation is based on discussions and experiences we made together with colleagues and partners in various projects over the last two years. Those projects dealt with different application domains but always offered a 2D space to facilitate a information organization and retrieval process and thus came with similar issues and challenges. Most of the observations were made during development by our team members and university colleagues. Industry partners were involved in two recent projects, giving helpful feedback and user scenarios (cf. Sec. 4 and [27]). The observations contributed to the planning of the next steps, including user studies to survey the impact of such a system on information retrieval processes compared to others.

5.1 Data matters

A key question of the work is about the semantic positioning of information entities in a 2D space and their reaction to user input. Staying in the domain of films and actors, when a node keeps a greater distance to another than a user would expect, it is not clear to him if the positioning is bad/wrong or the system is just assuming a lower relevance than he does. As the quality of the knowledge base has a similar impact on the representation as the positioning system, may it be physics based or not, we had a hard time to tune the parameters. Every once in a while, when the knowledge base has grown, the link prediction algorithm or relevance calculation changed, the physics had to be adapted as well. Two findings in this regard:

(1) It is impossible to shrink "relevance" to a generic numerical value. For the positioning we argued, that context matters, the visual structure matters and the same is true for a relevance calculation between information entities.
(2) Beside the isolated values, their distribution is important. The shown calculation of an ideal distance (cf. Sect. 3.2) is a linear function, because it implies that relevance by proximity is interpreted in a linear way. However, if the distribution of the relevance value is not linear, but, e.g., normal, this leads to an uneven usage of the available space and most of the suggestions with a relevance of around 0.5 could hardly be distinguished.

An other concern is the quantity of information entities and relationships in the knowledge base. For the Demonstrator in Sect. 4, the database manages around 6,000 entities (movies, actors, etc.) and 35,000 relationships among them. The quantity influences the duration that is needed to process queries, which can be rather complex and requested multiple times in a second. If the duration the knowledge base needs to answer those queries is too long, the application does not feel interactive anymore. Furthermore we have to consider the limited available space to show suggestions. It may be sufficient for 10, 20, or 30 of them, but this can only be scaled up to a certain degree, if the application should show more than particles. Showing, e.g., 20 of 1000 relevant suggestions might be too fuzzy to be helpful.

5.2 Physics playground

The advantage of using a physics simulation for visualization purposes is shown in Sect. 3.1. Another strength is, that interacting with physics makes fun. Moving nodes around, watching the reaction of the suggestions and the process of how they organize themselves in the space feels like playing a game. A good comparison is the game "Goat Simulator", whose main goal is to do funny things with a goat in a 3D world, simulated by ragdoll physics. This is not just an end in itself, it fosters creativity by motivating users to try out different combinations of structures and keywords, but takes away attention that might be focused on examining the actual suggestions, not just there simulated behavior.

In most situations the simulation leads to comprehensible positions and transitions. As the system is self-organizing over time, it is not necessary to implement special animations for those. A downside of this is a sluggish response to interactions. We argued, that the movement of objects needs to be damped, otherwise suggestions would behave very volatile and erratic, making it impossible to trace what is happening on the screen[2]. It is possible to tweak the objects' properties, but it turned out that physics simulations can behave unexpected in many cases. Especially when a lot of objects are involved it is hard to produce or prevent a certain behavior.

When too many suggestions are part of the simulation, physics can get confusing, because the paradigm implies that each object influences each other, may it be by forces of repulsion or attraction of the spring model, mutual repulsion of suggestions and collisions. The latter can result in suggestions jamming each other, because they are attracted in different directions, but collide and thus block each other. Fur sure, it would be possible to fiddle around, e.g., by implementing suggestions as circular objects or to increase the general force of repulsion. But everything that can be done, might have another negative impact. Circular objects occupy more space compared to rectangular ones, when content of suggestions is displayed as text, an increased force of repulsion influences distance and thus the semantics (cf. Sect. 3). When the number of suggestions exceed that of user nodes by far (roughly a factor greater than ten), the connections starting at the user nodes getting more and more irrelevant as they contribute less to the resulting total force.

5.3 Pitfalls

While physics simulation solves most problems of positioning and interaction, some of them are out of its scope. Whenever multidimensional information is displayed in a space of less dimensions get lost. Often, this is not an issue, since it is sufficient to find a solution that is good enough. In some cases this does not work, because the result cannot be interpreted correctly. In Fig. 6 such a case is shown in the upper half. It is impossible to find a position for the suggestion, that is close to nodes A, B, and D, but far away from node C. The simulation calculates a position with the lowest potential energy in the system possible, thus the suggestion is too close to C – implying a high relevance – and too far from D. A possibility to improve those cases is to show suggestions more than once, also depicted in the lower half of Fig. 6. Instead of representing the suggestions as one object, it is split to an object (a) and (b). The latter is decoupled from nodes A and B, allowing it to float near D and still keeping a suitable distance to C. Accordingly, (a) is not influenced by D. A simple physics model is not aware of such and

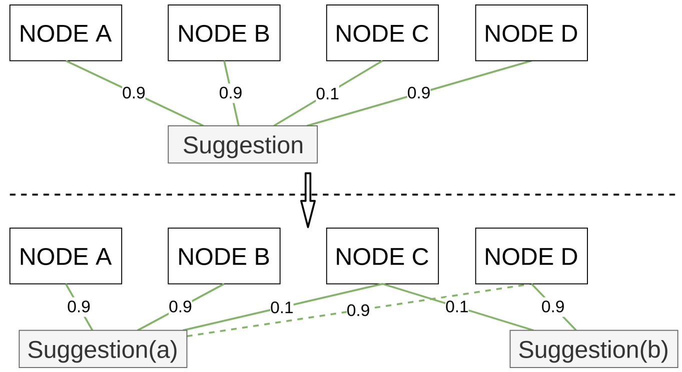


Figure 6: The upper half shows a suggestion positioned by a spring model. In the lower half the suggestion is split in two parts, to improve visual representation of relevance.

similar semantic misinterpretations, thus additional algorithms, e.g., the described one, are needed.

Currently, the spatial parser generates queries by identifying visual related nodes to find suggestions that are relevant to those. A user is moving nodes, making such clusters bigger or smaller and often, two clusters of visual related nodes unify to one. At the moment of writing, the demonstrator deletes all former suggestion and adds those that are part of the query result. Obviously, the unified result is strongly related to the results before the clusters merged. To redraw all of the suggestions seems to be a bad solution, because we demanded traceability of all system induced actions, especially when redrawing objects that were already part of the space. As described before, a physics simulation cannot handle this on its own. A potential solution would be to keep objects that would be redrawn, but adapt the distance joints. If a suggestion appeared for both original clusters, they could fuse into one or used as a split suggestion.

6 CONCLUSION AND FUTURE WORK

Altogether this paper describes the problem of representing semantics in a 2D space and the benefit of creating queries visually. The physics simulation underlying model is a starting point for further research effort in the field of visualizing semantics as human beings would do. It is a novel approach to utilize the recognition of visual structures and realizing visualization of the results as manipulation of the same spatial hypertext the users are using for organizing their thoughts and intends. The recognition of visual structure influences the queries sent to the knowledge base, thus a query can be refined without changing the terms forming it. It is a fundamental principle in the system's design that objects in space do represent more than a simple keyword to offer a unified environment to consume, organize, search, or retrieve information. The applied physics model, an adapted spring metaphor, helps to position or manipulate system generated suggestions in such a way that human recipients are able to interpret relations between user or suggestion nodes appropriately. By adding or discarding suggestions users refine their search without typing or the need of prior knowledge.

[2]See the demo video on YouTube: https://youtu.be/GX53yezHDXE

In Sect. 5 we have shown open issues of the current system. While some of them can be addressed with additional algorithms, others are an inevitable effect of using physics. Therefore we will improve the current implementation and examine further possibilities to achieve similar or better results. For example it is worth a while to put some work into the early approaches (cf. Sect. 3.1) to find solutions to support a descent behavior when users manipulate objects' visual properties. The physics simulation encodes relevance by proximity and behavior only, but there are further visual properties that can be used, e.g., color, shape, size, or transparency.

A generic approach of assigning relevance between pairs of information entities is followed. Any such relation can be expressed as a value. Those serve as the main parameters for the visualization. We consider a whole network of such related entities. Let's assume two nodes, A and B, and a suggestion S, which is very relevant to A, but not at all to B. What is the "combined relevance" of S to A and B? It may be something like an average value, but it can very well be, that the existence of A changes the relevance of S to B. A and B are part of the same query because the user put them into a visual relationship for a reason. The spatial parser recognizes this fact, the knowledge base should react to it as well by calculating dynamic, context aware values on demand.

Spatial parsers identify way more nuances of visual relationships than the binary choice between related and not related does indicate. We clearly can see a stronger visual relation when two nodes are close to each other and share the same color, compared to a situation where they have a completely different hue. This might be an information of interest when examining clusters of three and more nodes, because – like described in the paragraph above – this contextual information is given by the user. The limited available space to show suggestions demands the retrieval of the most important suggestions, not just a good positioning of those.

The presented work helped in shaping a concept that answers a number of questions related to pre-existing applications or techniques. The next big step is to implement user studies to quantify observations we made. They will help us to understand how users will work with such a system, which visual clues they prefer, the accepted degree of uncertainty, or which specific values should be used, e.g., for proximity, to encode a certain contextualized relevance.

This work will find its application in a variety of upcoming projects. It is an essential part of our approach of connecting humans' intuitions, creativity, or tacit knowledge with machines' capabilities of processing vast amounts of data. As such, any application domain that includes creative tasks and benefits from machine intelligence is a good candidate for our approach [cf. 2]. It combines Doug Engelbart's original idea of "augmenting human intellect" [9] with today's advances in AI. With systems similar to the proposed one we will reach a higher level of quality, effectiveness, and efficiency in problem solving compared to human-only or machine-only approaches. We expect this research direction to become the next "synchronization point", as described in our 2017 paper [4] and to gain in importance among the discussions within the hypertext community.

ACKNOWLEDGMENTS

The financial support by the Bavarian State Ministry of Science and the Arts (grant ID 1547-RP-01) and the TechnologieAllianzOberfranken (TAO) is gratefully acknowledged.

REFERENCES

[1] Jae-wook Ahn and Peter Btusilovsky. 2009. Adaptive Visualization of Search Results: Bringing User Models to Visual Analytics. *Information Visualization* 8, 3 (2009), 167–179. https://doi.org/10.1057/ivs.2009.12

[2] Claus Atzenbeck and Peter Nürnberg. 2019. Hypertext as a Method. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media (HT '19)*. ACM. In print.

[3] C. Atzenbeck, D. Roßner, and M. Tzagarakis. 2018. Mother - An integrated approach to hypertext domains. In *HT 2018 - Proceedings of the 29th ACM Conference on Hypertext and Social Media*. https://doi.org/10.1145/3209542.3209570

[4] Claus Atzenbeck, Thomas Schedel, Manolis Tzagarakis, Daniel Roßner, and Lucas Mages. 2017. Revisiting Hypertext Infrastructure. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media (HT '17)*. ACM, New York, NY, USA, 35–44. https://doi.org/10.1145/3078714.3078718

[5] Christoph Beckmann and Tom Gross. 2010. Towards a group recommender process model for ad-hoc groups and on-demand recommendations. In *Proceedings of the 16th ACM international conference on Supporting group work - GROUP '10*. ACM Press, New York, New York, USA, 329. https://doi.org/10.1145/1880071.1880134

[6] Mark Bernstein. 2011. Can we talk about spatial hypertext. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia - HT '11*. ACM Press, New York, New York, USA, 103. https://doi.org/10.1145/1995966.1995983

[7] Matthew Chalmers and Paul Chitson. 1992. Bead: Explorations in Information Visualization. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '92)*. ACM, New York, NY, USA, 330–337. https://doi.org/10.1145/133160.133215

[8] Paul Dourish and Matthew Chalmers. 1994. Running out of space: Models of information navigation. In *Short paper presented at HCI*, Vol. 94. 23–26.

[9] Douglas C. Engelbart. 1962. *Augmenting Human Intellect: A Conceptual Framework*. Summary Report AFOSR-3233. Standford Research Institute. http://dougengelbart.org/content/view/138

[10] Walter C. Gogel. 1963. The visual perception of size and distance. *Vision Research* 3, 3-4 (mar 1963), 101–120. https://doi.org/10.1016/0042-6989(63)90033-6

[11] Tom Gross. 2015. Supporting Informed Negotiation Processes in Group Recommender Systems. *icom* 14, 1 (jan 2015), 53–61. https://doi.org/10.1515/icom-2015-0008

[12] Frank G. Halasz. 1987. Reflections on NoteCards: Seven issues for the next generation of hypermedia systems. In *Proceedings of the ACM Conference on Hypertext*. ACM, 345–365. http://doi.acm.org/10.1145/317426.317451

[13] Eric Horvitz. 2003. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99*. ACM Press, New York, New York, USA, 159–166. https://doi.org/10.1145/302979.303030

[14] T. Igarashi, S. Matsuoka, and T. Masui. 1995. Adaptive recognition of implicit structures in human-organized layouts. *Proceedings of Symposium on Visual Languages* (1995), 258–266. https://doi.org/10.1109/vl.1995.520817

[15] Khalil Klouche, Tuukka Ruotsalo, Luana Micallef, Salvatore Andolina, and Giulio Jacucci. 2017. Visual Re-Ranking for Multi-Aspect Information Retrieval. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval (CHIIR '17)*. ACM, New York, NY, USA, 57–66. https://doi.org/10.1145/3020165.3020174

[16] Stephen G. Kobourov. 2012. Spring Embedders and Force Directed Graph Drawing Algorithms. (jan 2012). arXiv:1201.3011 http://arxiv.org/abs/1201.3011

[17] Uwe Krohn. 1996. VINETA: navigation through virtual information spaces. In *Proceedings of the workshop on Advanced visual interfaces - AVI '96*. ACM Press, New York, New York, USA, 49. https://doi.org/10.1145/948449.948458

[18] Jock Mackinlay. 1986. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics* 5, 2 (apr 1986), 110–141. https://doi.org/10.1145/22949.22950

[19] C.C. Marshall and F.M. Shipman III. 1993. Searching for the missing link: discovering implicit structure in spatial hypertext. *Proceedings of the fifth ACM conference on Hypertext* November (1993), 217–230. http://portal.acm.org/citation.cfm?id=168826

[20] Catherine C Marshall, Frank G Halasz, Russell A Rogers, and William C Janssen Jr. 1991. Aquanet: A Hypertext Tool to Hold Your Knowledge in Place. In *Proceedings of the Third Annual ACM Conference on Hypertext (HYPERTEXT '91)*. ACM, New York, NY, USA, 261–275. https://doi.org/10.1145/122974.123000

[21] Catherine C Marshall and Frank M Shipman III. 1995. Spatial Hypertext: Designing for Change. *Commun. ACM* 38, 8 (aug 1995), 88–97. https://doi.org/10.1145/208344.208350

[22] Catherine C Marshall, Frank M Shipman III, and James H Coombs. 1994. VIKI: Spatial Hypertext Supporting Emergent Structure. In *Proceedings of the 1994 ACM European Conference on Hypermedia Technology (ECHT '94)*. ACM, New York, NY, USA, 13–23. https://doi.org/10.1145/192757.192759

[23] A Nuchprayoon and R R Korfhage. 1994. GUIDO, a visual tool for retrieving documents. In *Proceedings of 1994 IEEE Symposium on Visual Languages*. 64–71. https://doi.org/10.1109/VL.1994.363639

[24] Kai Olsen, James G Williams, Kenneth M Sochats, and Stephen Hirtle. 1991. Ideation through visualization: The VIBE System. *Multimedia Review* 3 (1991).

[25] Olav Reinert, Dirk Bucka-Lassen, Claus Aagaard Pedersen, and Peter J Nürnberg. 1999. CAOS: A Collaborative and Open Spatial Structure Service Component with Incremental Spatial Parsing. In *Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia : Returning to Our Diverse Roots: Returning to Our Diverse Roots (HYPERTEXT '99)*. ACM, New York, NY, USA, 49–50. https://doi.org/10.1145/294469.294484

[26] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). 2011. *Recommender Systems Handbook*. Springer US, Boston, MA. https://doi.org/10.1007/978-0-387-85820-3

[27] D. Roßner and C. Atzenbeck. 2018. Spatial hypertext for end-user development tools. In *HUMAN 2018 - Proceedings of the 1st Workshop on Human Factors in Hypertext, Co-located with HT 2018*. https://doi.org/10.1145/3215611.3215612

[28] Thomas Schedel. 2017. Spatio-Temporal parsing in spatial hypermedia. In *ACM SIGWEB Newsletter (HT '16)*. Aalborg University, ACM, New York, NY, USA, 1–5. https://doi.org/10.1145/3027141.3027142

[29] Frank M. Shipman, Robert Airhart, Haowei Hsieh, Preetam Maloor, J. Michael Moore, and Divya Shah. 2001. Visual and spatial communication and task organization using the Visual Knowledge Builder. In *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*. ACM Press, 260–269. http://doi.acm.org/10.1145/500286.500325

[30] Frank M Shipman III, Catherine C Marshall, and Thomas P Moran. 1995. Finding and Using Implicit Structure in Human-organized Spatial Layouts of Information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 346–353. https://doi.org/10.1145/223904.223949

[31] Anselm Spoerri. 1993. Infocrystal: A Visual Tool for Information Retrieval. In *Proceedings of the 4th Conference on Visualization '93 (VIS '93)*. IEEE Computer Society, Washington, DC, USA, 150–157. http://dl.acm.org/citation.cfm?id=949845.949876

[32] T.C. Sprenger, M.H. Gross, D. Bielser, and T. Strasser. 1998. IVORY-an object-oriented framework for physics-based information visualization in Java. In *Proceedings IEEE Symposium on Information Visualization (Cat. No.98TB100258)*. IEEE Comput. Soc, 79–86,. https://doi.org/10.1109/INFVIS.1998.729562

[33] Jin Zhang. 2001. TOFIR: A tool of facilitating information retrieval - Introduce a visual retrieval model. *Information Processing and Management* 37, 4 (2001), 639–657. https://doi.org/10.1016/S0306-4573(00)00042-X

[34] Jin Zhang and Robert R Korfhage. 1999. DARE: Distance and angle retrieval environment: A tale of the two measures. *Journal of the American Society for Information Science* 50, 9 (1999), 779–787. https://doi.org/10.1002/(SICI)1097-4571(1999)50:9<779::AID-ASI6>3.0.CO;2-R