User-Supported Interactive Illustration of Text

Marcel Götze*
Otto-von-Guericke Universität Magdeburg, Germany

Petra Neumann, Tobias Isenberg†
University of Calgary, Canada

Abstract

This paper presents a novel way of illustrating textual documents. If one lacks the ability, time, or skill to create illustrations by hand he or she can use a 3D model for the creation or search for an appropriate image. By having a set of 3D models from an appropriate subject matter, we implemented a flexible system that allows it to control the final imagery, both in terms of viewing angle and size and in terms of visual style. The style of rendition may be adapted to the text by using photorealistic, non-photorealistic, and hybrid rendering. For the search for an appropriate image, numerous sources are available, e. g., image databases or the World Wide Web. Therefore, the process of illustrating a text has shifted from the creation of the image to the search for the right image. We present a method that supports the interactive search for a 3D model that can also be used to find illustrative images using a Web search engine. The search can be performed interactively or automatically and is based on keywords. Embedding an image into the text is simplified to a drag and drop operation from the search results into the text while figure caption and figure references are automatically generated, based on the selected keywords.

1 Introduction

Since the early days of making books, authors have added images to their texts to explain things, to clarify certain aspects using a different medium (the image), or just to make reading a little more amusing. The underlying problem motivating our research is almost as old as book making itself—finding or creating images that serve as illustrations and embedding these images in a given text. Solutions to this problem have changed over time. In the early days, all illustrations had to be created by the author or a skilled illustrator. Today, authors can, on the one hand, use 3D models of objects to create an image. For this purpose different rendering styles have been developed in the past. Depending on the communicative goal the author can choose between photorealistic (PR) and non-photorealistic (NPR) techniques. Furthermore, both can be combined to create hybrid renditions. The problem in this case is to find the right model within the database of 3D models. On the other hand, image databases and the WWW (e. g., Web search engines such as Altavista, Google, or AlltheWeb) provide an unmanageable amount of images that can be included in a publication. In most cases the information needed to find and access a certain image is simply a set of keywords.

Putting it all together, authors have applications at their disposal that provide functionality to search for images and others that provide functionality to include images in texts. The illustration problem today, thus, can be decomposed into the following steps:

[*] Fakultät für Informatik, Institut für Simulation and Graphik, D-39016 Magdeburg, Germany

[†] Department of Computer Science, Calgary, Alberta, Canada T2N 1N4

1. examine the parts of the text where an illustration should be placed,
2. find the (key)words explaining the illustration,
3. access a database of 3D models or point a (Web) browser to a search engine an image database,
4. use the keywords to perform a query,
5. find the right image in the search results or render an illustration from a located 3D model,
6. embed the image in the text and add a caption, and,
7. reference the illustration in the text.

In this paper we present a method for illustration that combines the functionality of a word processing application and an application to search for digital images or models that can be used to illustrate pages of a textbook. For the creation of an image based on a 3D model we developed a rendering system that can create photorealistc illustrations as well as non-photorealistic renditions.

2 Related Work

The work presented in this paper bridges the gap between the work with textual documents and image search. In between, topics like annotations as well as information seeking, illustrations in electronic documents, illustration using different rendering styles, the creation of figure captions, and interaction with illustrations are being touched.

The first topic to investigate is annotating a document. In our context images are also to be regarded as annotations attached to parts (words) of the document. PHELPS and WILENSKY categorize and describe different types of user manipulations for annotating a document [PW97]. The authors do not directly introduce images as annotations but beside many other types they present freehand sketches that come close to images.

Another interesting aspect is the use of handdrawn annotations for information seeking as introduced by GOLOVCHINSKY, PRICE and SCHILIT in [GPS99, PGS98]. The authors introduce the use of freehand annotations to retrieve documents for further reading. The proposed method is not explicitly used to find images for illustration but, nevertheless, it is inspiring for our work and touches parts of it.

As mentioned above for the illustration of a text an author can use static images resulting from a Web based image search as well as images created based on a 3D model. Related to the search for static images, numerous publications are available dealing with the collection and indexing of images which applies more to data mining than to the work presented here. However, some applications in this area are inspiring especially for the use of image-databases. For example, in [SFA96] and [Scl95] new types of search engines are presented that not only analyze the text on the image's Web page but also the image itself. Furthermore, in [ARS98] an approach for image search is introduced that is based on a multimedia model to describe different multimedia components. The methods introduced here are helpful for the design of the database search module.

CHEN et al. describe iFind [CWH$^+$01], an application to find images based on both a textual description (keywords) and image examples as well as a combination of both methods. Nevertheless, this program is not designed to support illustration, it only shows search results. A distributed way of image seeking is introduced by SCLAROFF, TAYCHER, and CASCIA in [STC97]. Their system, ImageRover, uses a set of distributed robots to gather images from web pages. The interesting point is the novel relevance feedback algorithm that allows the user to choose the most relevant images and the system to compute new queries based on calculations using distance metrics.

LIEBERMAN and LIU introduced in [LL02a, LL02b] a method that uses world semantics for image retrieval. This technique makes information retrieval more robust than the use of keywords. But since our illustration method is more tailored to the simplification of the interaction with the system we use the keyword search as a first approach. For further development it would also be interesting to use different methods for the analysis of the text.

In the following we will discuss work related to the creation of images for illustration based on 3D models using different rendering styles. In recent years, there have been a great number of publications on various styles of non-photorealistic rendering (for an overview see, e. g., [GG01] or [SS02]). Among these, there have been a considerable number that specifically deal with the creation of NPR illustrations. Insights into the topic of scientific illustration using specifically NPR approaches are given, for example, by SOUSA [Sou03] and EBERT [Ebe03]. They describe the process of creating an illustration using non-photorealistic rendering techniques from the technical as well as from the perceptional point of view. This has a great impact for the development of NPR visualization applications. With respect to the use of de-emphasis in illustration, DIEPSTRATEN et al. discuss how to efficiently render transparency in NPR illustrations [DWE02]. They introduce a new view-dependent transparency model and show its hardware-accelerated implementation. In addition to the few examples named above, there are many more (also, e. g., in other areas of NPR such as volume rendering) but a complete review would be beyond the scope of this section.

Finally, the combination of images and text plays an important role for the presented research. SCHLECHTWEG and STROTHOTTE introduce an approach that on the one hand allows it to change the visualization of a 3D model by interacting with the text. On the other hand it shows that interaction with the 3D model can be used to influence the visualization of the text [SS99]. The authors do not analyze the text over and above the extraction of keywords. An extension to this work is presented by HARTMANN et al. [HSHS02] where a knowledge base containing domain knowledge is exploited to couple images and text in an interactive system. A special area within this realm is the creation of figure captions for images that were rendered from a 3D model. PREIM et al. developed a method to automatically generate figure captions based on the visualization of the 3D model [PHS98]. Information about the position and transformations of the 3D model as well as the viewer position are used to fill in slots in a figure caption template to finally describe what is shown in the image and how the parts of the model are positioned.

3 Illustrating by Marking

For being able to efficiently integrate illustrations into a text a user has to acquire the illustrating object first. For this purpose we developed SEARCHILLUSTRATION, which is the foundation for both types of illustrations discussed later on. It describes the notion that the application searches for illustrations rather than the user. In the following we will describe the concept and the components that are needed to achieve this goal. In general, at least two parts are required in an interactive illustration application: a text editor and a repository of illustration objects together with an API to perform a keyword based search for them.

While the text editor is self-explanatory, the second point needs a more detailed examination since several alternatives exist. We will show two of them in more detail. First, we introduce the illustration using static images resulting from an image search engine on the World Wide Web since this is a widely available option. Second, we will show how a 3D model database can be used in our context to create photorealistic and non-photorealistic illustrations that—in contrast to existing digital images—can even be customized.

3.1 General Search Method

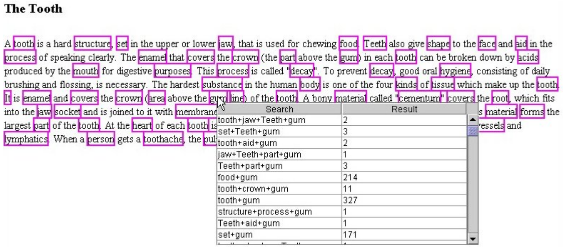
The general illustration procedure resembles the steps mentioned in the introduction. The main difference is that the user does no longer need to take care of finding keywords, transferring them to a search interface, thereby initiating the image search. While writing, keywords for a query string are extracted from the text—either automatically or manually—and sent to the respective search interface. The search results are generated and presented to the user so that he or she can select the image that best fits the illustrative needs at hand.

3.2 Query String Generation

The image search that needs to be performed is driven by a query string that basically consists of a list of keywords. To find an illustration for a given text, these keywords usually contain the main nouns from the text. The query string itself can be generated in two different ways, automatically or interactively. An automatic query string generation hides the process completely from the user while a manual query string generation gives the user somewhat more control. Our results obtained with a prototypical system showed that both variants have their rightful place and can be applied depending on the user's needs and experience.

3.2.1 Automatic generation of query strings

All the information needed to find an image in a collection of images can be extracted from the text. A text analysis procedure that uses a part of speech tagger (as, for example, introduced in [TM98]) collects all nouns that have been typed in by the user.



The Tooth

A tooth is a hard structure, set in the upper or lower jaw, that is used for chewing food. Teeth also give shape to the face and aid in the
process of speaking clearly. The enamel that covers the crown (the part above the gum) in each tooth can be broken down by acids
produced by the mouth for digestive purposes. This process is called "decay". To prevent decay, good oral hygiene, consisting of daily
brushing and flossing, is necessary. The hardest substance in the human body is one of the four kinds of tissue which make up the tooth.
It is enamel and covers the crown (area above the gum line) of the tooth. A bony material called "cementum" covers the root, which fits
into the jaw socket and is joined to it with membrane ... s material forms the
largest part of the tooth. At the heart of each tooth is ... essels and
lymphatics. When a person gets a toothache, the pul...

Search	Result
tooth+jaw+Teeth+gum	2
set+Teeth+gum	3
tooth+aid+gum	2
jaw+Teeth+part+gum	1
Teeth+part+gum	3
food+gum	214
tooth+crown+gum	11
tooth+gum	327
structure+process+gum	1
Teeth+aid+gum	1
set+gum	171

Figure 1: Automatic selection of nouns in the text (boxed) leads to different query strings. All successful queries are shown in a table for selecting the most appropriate one.

This text analysis runs permanently in the background. For each newly found noun new query strings are generated. Given n nouns, all possible combinations of i ($i \leq n$) out of n nouns are generated as query strings. This way, the number of queries generated grows quickly (considering all possibilities, $2^n - 1$ queries would be generated given n nouns) so that a limitation is needed. The search is initiated immediately and unsuccessful queries are eliminated. New query strings are now only generated from the set of successful queries. Our empirical studies show that this procedure is feasible since not all word combinations return a search result (cf. Figure 1).

3.2.2 Interactive generation of query strings

The user may also generate query strings interactively since the automatic search runs permanently while typing and a list of search results is presented whenever the user wishes. This can easily be done by marking the word using a pointing device (e. g. mouse or pen). Every new marking generates a new query string by adding the new word to the previously selected words. Since authors put more consideration in the selection of query terms than an automatic process would do it is reasonable to offer choices of how to proceed with the set of selected words. First of all, this set can be used as a single query similar to a sequence of words typed into a database's or search engine's search interface. Second, it can be used as a refinement to a previous query, i. e., the search is performed on the results of a previous query. A final option is to search for all combinations similar to an automatically generated query string.

4 Obtaining Images for Text Illustration

Once the user has identified the part that has to be illustrated and selected keywords for a query he or she needs to obtain the right images from an image source. Our system offers two choices, selecting images from the World Wide Web or choosing and creating images from models loaded from a 3D model database. Both choices will be described in this section.

4.1 Web based image retrieval

The actual query being submitted to an image database depends on the interface offered by that specific database. While special purpose image databases may work very well for texts with specific topics, they are expensive to obtain and to maintain. Many World Wide Web search engines, on the other hand, offer an image search free of charge. The search is performed on the body of all images being indexed by that particular search engine. There is another advantage when using web search for our purposes. The query itself can be encoded in a URL and sent to the search engine using the HTTP protocol. The results obtained are HTML pages that can be shown to the author for the selection of the image he or she wants to include.

Even though many Web search engines do offer an API, these interfaces are by no means standardized. Also, they do not always include image search features. Therefore, URL encoding offers the most flexible way to issue a query to any search engine if the specifics of that engine in terms of the format of the query URL are considered. Figure 5 shows a schematic overview of the system.

4.2 Illustrating with Photorealistic Images

Images obtained from a Web search are static images and their contents can hardly be influenced by the author. In many cases these images serve the purpose of illustrations well. If, however, the author wants very specific images then they may not be appropriate. In the following we will describe an alternative to the Web search discussed above where the user may influence the contents and design of the illustration he or she gets from a search. Instead of searching an image database or the Web we use a database of 3D models and generate the images from these models. It provides more flexibility for illustrating scientific and technical texts and creates unique images. Assume that a database of 3D models is available and that each model can be accessed using keywords. These keywords usually label parts of the model, i. e., single objects in a more complex model. Using these keywords the same search metaphor as mentioned above can be applied. As an example, the model of a human skull may contain parts labeled jaw, teeth, gum, etc. Using a standard 3D environment and an appropriate interface a user is now able to adapt the view on the scene and render an illustration. In order to enhance the illustration, different rendering styles can be used to

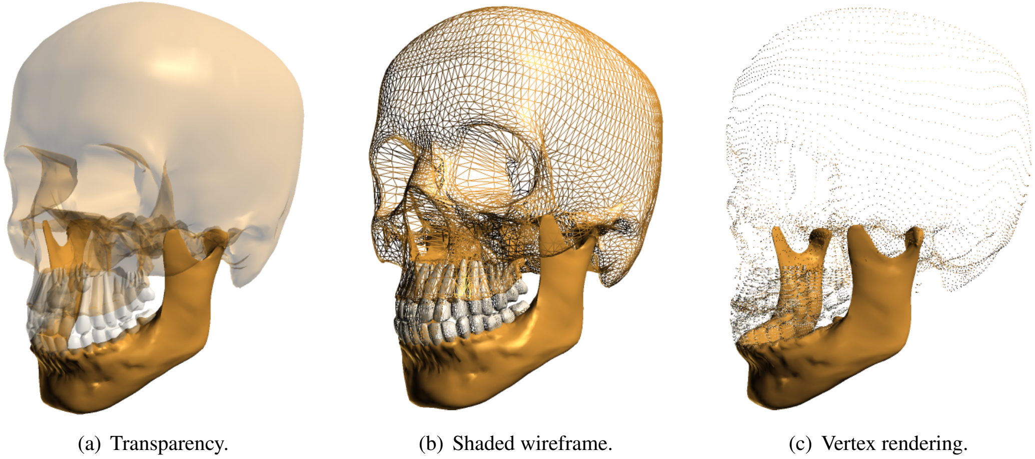

(a) Transparency.
(b) Shaded wireframe.
(c) Vertex rendering.

Figure 2: De-emphasis using photorealistic methods.

highlight the queried parts in the rendition. De-emphasis in these photorealistic renditions may be achieved, for example, by using transparency, vertex rendering, or wireframe rendering (see examples in Figure 2). Unfortunately, sometimes the detail present in photorealistic renditions as well as the characteristics of the rendering styles such as shading and highlights may distract from important aspects and create visual clutter. They may negatively affect the message that was intended by the illustration. In addition, in scenes with emphasized parts sometimes objects cannot easily be visually separated because there is no clear border between emphasized and de-emphasized objects (in particular, see Figures 2(a) and 2(b)).

On the other hand, depending on the domain there are situations where photorealistic renditions are well suited as illustrations. In fact, in some cases it is absolutely necessary to see how an object looks like in reality (i. e. in the field of architecture, archaeology, design, or for biological objects).

4.3 Illustrating with Non-Photorealistic and Hybrid Images

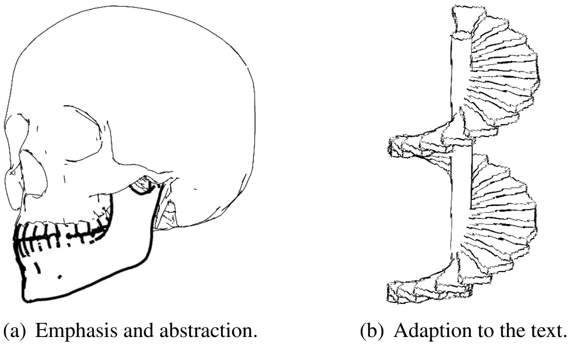
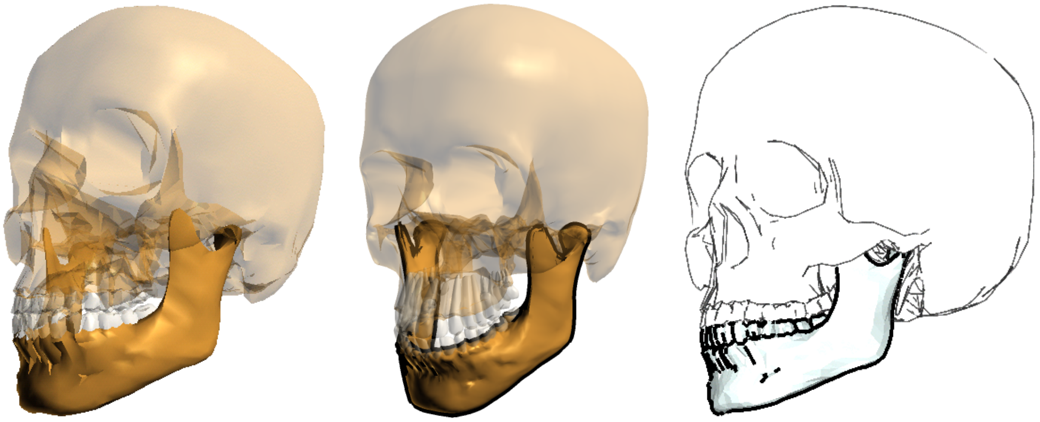

(a) Emphasis and abstraction.
(b) Adaption to the text.

Figure 3: Line styles emphasizing parts of a model and adapting the style of the illustration to the type of text.

In contrast to the previously discussed photorealistic methods, non-photorealistic rendering (NPR) aims at creating expressive and artistic renditions. A great number of styles have been conceived, many of which are well suited for creating illustrations. In particular, stroke based rendering styles are often used for illustrations (cf. Figure 3). These styles allow a very easy and effective stylization as well as abstraction. In addition, line drawings have been used for illustrating texts for a long time so viewers are used to this type of presentation. The wide range of possible line styles not only allows to emphasize the objects in the model that are referred to in the text but also makes it possible to adapt the choice of line style to the specific text. For example, in a technical documentation a clear style that generates straight lines should be employed while in texts about architectural sketches a more random line style may be used (see example in Figure 3(b)).

In addition to pure photorealistic or non-photorealistic rendering it is also possible to combine both styles to create hybrid renditions. As has previously been discussed by JESSE and ISENBERG, this extends the set of expression possibilities for a specific model in a visualization [JI03]. In these hybrid renditions it is possible to use the advantages of either rendering style. E. g., silhouette lines may be added to a photorealistically rendered model or to parts of it in order to emphasize objects or to enhance the borders of the objects (cf. Figure 4(b)). Different NPR methods can be combined and applied to different or the same objects in one rendition (cf. Figure 4(c)). Combining different visualizations of the same model can be employed to visualize different aspects of the same model. Therefore, these different visualizations can be used in different parts of the text.



(a) Transparency.

(b) Added silhouettes.

(c) Hybrid rendition.

Figure 4: Comparison of emphasis with transparency, with transparency and added silhouettes, and with hybrid rendering using non-photorealistic rendering styles combined with a shading. It can be observed than the latter emphasizes the important object with less visual disturbance that in (a).

4.4 Interface to Graphics Systems

To enable a text editor to communicate with a 3D model database a specific interface is needed. We decided to implement a client-server application to be able to adapt to different rendering applications and operating systems. The layout of our system includes four basic components, a text editor which includes the text to be illustrated, a 3D model database and the two interfaces to the graphics systems.

Figure 5 shows a schematic overview of the system. It shows that the tasks of creating the two different types of illustrations are handled by two separate graphics systems. The component for generating photorealistic images is based on traditional OpenGL rendering while the other one interfaces to a non-photorealistic rendering system. These will now be described in more detail:

In order to be able to interactively create and work with the photorealistically shaded images, we decided to interface to a traditional OpenGL rendering system. Once the user has selected a model that resulted from the database query, the name is transferred to the OpenGL system, the model is loaded, and the containing objects are being analyzed for further processing.

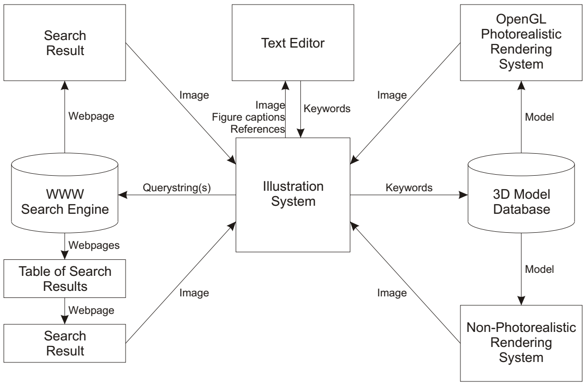


Figure 5: Schematic overview of the illustration system.

Today's graphics systems are capable of displaying most models in realtime. Therefore, the parameterization can be performed interactively by the user on the client side. In addition to changing the view onto the scene, parameters such as color, transparency, or photorealistic emphasis and deemphasis techniques may be used. For example, transparency can be applied to parts of the scene to emphasize the remaining objects as shown in Figure 2(a).

At any given time, the image generated can be included into the text for illustration, for example, by dragging it from the rendering window into the text. This way, the user is able to create the needed photorealistic images very fast and interactively illustrate the text. However, some artistic skill is required at this level. The user has to decide how to emphasize or deemphasize (parts of) the model. However, we believe the artistic skill needed is reduced to a minimum by typical illustation conventions used in different research areas as described above.

In order to utilize a non-photorealistic rendering, we opted to work with the existing system OpenNPAR [HIR^{+}03] rather than implement a new one. This way, we can make use of new techniques being added to the system almost immediately. To make the functionality provided by the rendering system available to a wider variety of applications including our own, we developed a simple client-server interface to the renderer.

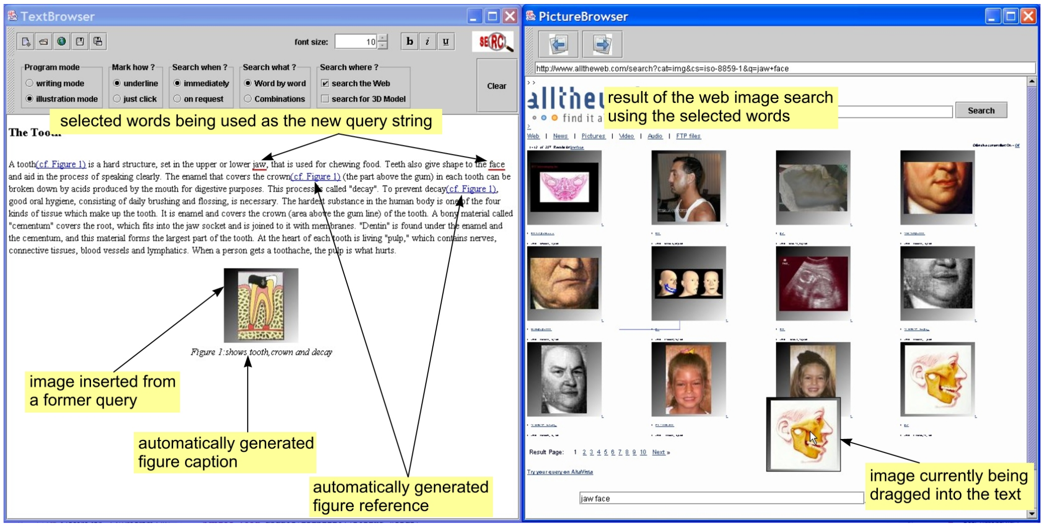
This client-server interface consists of a server on the side of the OpenNPAR system and a client on the side of the text illustration system. The server, a transaction processing module, receives commands or 3D geometry data from a connected client. Then, the module parses the received data and invokes corresponding commands within OpenNPAR or loads a received model file. After completing the rendition, the module takes the rendered image and sends it back to the client.

The client sends out information about the geometry to be loaded, the interaction that changes the view of the model (e. g., rotation etc.), and provides a user interface to the functionality provided by OpenNPAR. Each time a repaint is invoked on the server side and after this rendering has been completed, the client receives a single image from the OpenNPAR server.

4.5 Embedding the Images Into the Text

As described above, including the image into the text should be made simple for the user. We chose to offer a drag and drop mechanism.

Depending on the image source, different methods have to be invoked here. In case of a web search (cf. Figure 6), the image has to be loaded from the given URL; if the image is created from a 3D



Program mode: ○ writing mode ● illustration mode
Mark how ?: ● underline ○ just click
Search when ?: ● immediately ○ on request
Search what ?: ● Word by word ○ Combinations
Search where ?: ☑ search the Web □ search for 3D Model

The Tooth

A tooth(cf. Figure 1) is a hard structure, set in the upper or lower jaw, that is used for chewing food. Teeth also give shape to the face and aid in the process of speaking clearly. The enamel that covers the crown(cf. Figure 1) (the part above the gum) in each tooth can be broken down by acids produced by the mouth for digestive purposes. This process is called "decay". To prevent decay(cf. Figure 1), good oral hygiene, consisting of daily brushing and flossing, is necessary. The hardest substance in the human body is one of the four kinds of tissue which make up the tooth. It is enamel and covers the crown (area above the gum line) of the tooth. A bony material called "cementum" covers the root, which fits into the jaw socket and is joined to it with membranes. "Dentin" is found under the enamel and the cementum, and this material forms the largest part of the tooth. At the heart of each tooth is living "pulp," which contains nerves, connective tissues, blood vessels and lymphatics. When a person gets a toothache, the pulp is what hurts.

Figure 1:shows tooth,crown and decay

http://www.alltheweb.com/search?cat=img&cs=iso-8859-1&q=jaw+face

jaw face

selected words being used as the new query string

result of the web image search using the selected words

image inserted from a former query

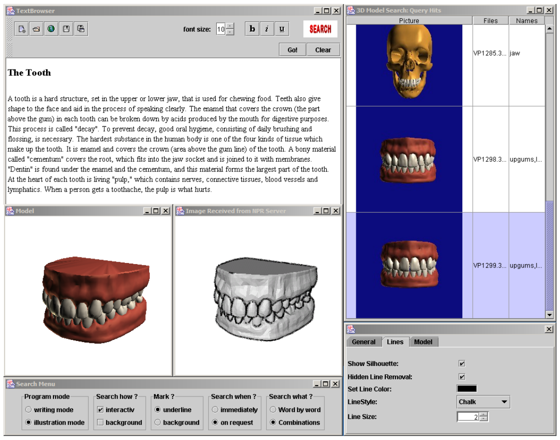
automatically generated figure caption

automatically generated figure reference

image currently being dragged into the text

Figure 6: Screenshot showing the embedding of images based on a web search.

model database a snapshot has to be taken (cf. Figure 7). In our approach this process has been hidden from the user to ease the illustration of a given text even more. Dropping an image into the text creates the need for placing it at an appropriate position. Here, rule-based algorithms can be employed that compute an optimal position of the image. For simplicity, in our prototype the image is just placed below the last paragraph of text containing the search terms. However, placing the image at the right place is a task that poses many problems in current text processing systems and remains an open problem.



The Tooth

A tooth is a hard structure, set in the upper or lower jaw, that is used for chewing food. Teeth also give shape to the face and aid in the process of speaking clearly. The enamel that covers the crown (the part above the gum) in each tooth can be broken down by acids produced by the mouth for digestive purposes. This process is called "decay". To prevent decay, good oral hygiene, consisting of daily brushing and flossing, is necessary. The hardest substance in the human body is one of the four kinds of tissue which make up the tooth. It is enamel and covers the crown (area above the gum line) of the tooth. A bony material called "cementum" covers the root, which fits into the jaw socket and is joined to it with membranes. "Dentin" is found under the enamel and the cementum, and this material forms the largest part of the tooth. At the heart of each tooth is living "pulp," which contains nerves, connective tissues, blood vessels and lymphatics. When a person gets a toothache, the pulp is what hurts.

Picture	Files	Names
	VP1285.3...	jaw
	VP1298.3...	upgums,l...
	VP1299.3...	upgums,l...

Program mode: ○ writing mode ● illustration mode
Search how ?: ☑ interactiv □ background
Mark ?: ● underline ○ background
Search when ?: ○ immediately ● on request
Search what ?: ○ Word by word ● Combinations

Show Silhouette: ☑
Hidden Line Removal: ☑
Set Line Color: ■
LineStyle: Chalk
Line Size: 2

Figure 7: Screenshot of the system showing the text window (top left), the result of the database query (top right), both visualizations of the chosen model (middle left), and the parameterization of the illustrations system (bottom left) as well as of the NPR visualization (bottom right).

After placing the image, a figure caption is created based on the selected keywords. A simple approach uses a template-based method where placeholders are filled with the respective search terms. In addition to the figure caption, a reference to the image is generated and placed within the text next to the selected search terms. This reference might also act as a hyperlink to the illustration and, thus, allows it to quickly navigate to the desired image.

5 User Interviews

We set up a test environment and performed a small informal user study (8 participants) to assess the feasibility of the approach and to see whether users are able to use the methods described so far in an appropriate manner. An introduction to the application was given to the participants and they were then asked to illustrate a given text. First, the illustration should be done using the interactive search feature. Then, the subjects were asked to illustrate the text based on a search using a 3D model database. The last task was to write a short paragraph of text while the system performs an automatic search. Afterwards, the participants were interviewed to the following topics:

- advantages of using SEARCHILLUSTRATION compared to the traditional way of illustrating a text using images from either the World Wide Web or image databases,
- various methods of generating query strings,
- automatic search,
- automatic image placement,
- creation of figure captions and references, and
- illustration using 3D models.

All participants rated the methods as very useful and appreciated the amount of time being saved in the illustration process. Many steps that are necessary for illustrating a text using a standard word processing application have been automated. Comparing the prototypical system to other applications, the overall assessment was that this method saves time - ”it is faster” - and makes the search for images easier.

Within the different methods to compute the query string when using the interactive search, the combination of search terms was rated positive because it saves the user from typing many different search strings. Furthermore, the set of computed combinations of search terms might also contain combinations that would possibly not have been considered by the user. In comparison to searching for just the words being marked in the text (the second possible way for interactive search) this results in more possible images. One subject argued that different combinations of search terms describe different images and so by using all possible combinations the probability of finding the right image is much higher.

The automatic generation of query strings and, hence, the automatic search was regarded as helpful and handy. However, many of the subjects would have preferred a better text analysis than just using all nouns as search terms.

The 3D model search has been assessed as very useful provided that the database is large enough. In this context the interaction was rated as good but some subjects suggested that it would be helpful if the visualization could be interactively parameterized to a larger extent (e. g., the color, the transparency or visibility of parts of the model).

The automatic creation of figure captions was judged as being very useful although the templates being used could be more sophisticated. The subjects highly appreciated the automatically generated figure references and the automatic figure numbering. Some subjects mentioned that they would in any case edit the (generated) figure caption but having the keywords describing the image is a good starting point for an individual figure caption. It also helps to keep in mind which keywords are described by the contents of the image.

In general the system was rated as easy to use and the possibility to add an image per drag and drop makes it even more helpful.

6 Concluding Remarks and Future Work

In this paper we present a novel way to illustrate textual documents that is based on a search for different images. The goal of this approach was to reduce the interaction steps to the absolutely necessary: marking the words that describe the image, choosing the right image, dragging and dropping it into the text. We showed that this approach can be used for different image sources ranging from search engines over image databases to databases of 3D models. Hence, this allows us on the on hand to use static images that can not be changed according to their communicative goal. On the other hand, the creation of images based on a 3D model that we demonstrated is individually adaptable.

We see many application domains, for instance journalism and education. Journalists usually have large image databases at their disposal to find images to be included in an article currently being written. Teachers who want to prepare handouts for students will most probably use the web image search feature instead of creating the images themselves.

There is, however, room for improvements. First of all, it should be thoroughly tested how the performance of the background search depends on various features like the number of query terms, the used search engine and the connection speed being available. There is one issue which definitely needs further attention when using SEARCHILLUSTRATION. Many of the images found via a Web image search are freely available but, nevertheless, the image source should be mentioned within the document that includes the image. For those images that are not freely available, permissions to copy these images have to be obtained. Both aspects can be supported by SEARCHILLUSTRATION when the image source is stored together with the image or document. The URL of the Web page where the image is taken from can, for example, automatically be included in the figure caption or in a list of figures. Such a list can also serve as a reference to ease the process of obtaining publication rights.

The general idea behind SEARCHILLUSTRATION can be extended to handle other types of information, such as statistical data, examples or just interesting facts. This means that based on the presented method different sources of information can be queried to provide the user with different kinds of data to be included in the document.

When using SEARCHILLUSTRATION together with a 3D model database, the security aspects are also interesting to describe the created illustrations. All information concerning the created image (keywords, 3D parameters, etc.) are available and could be embedded in the image either using watermarks or other description techniques for images. The placement of images can be improved by further parameterization (e. g., in rows, columns, etc.). In particular for the use in non-photorealistic illustrations, vector graphic support will be added to support a higher quality of the reproduction of the printed text.

References

[ARS98] Giuseppe Amato, Fausto Rabitti und Pasquale Savino. *Supporting Image Search on the Web.* In: The Challenge of Image Retrieval – Proc. of Workshop on Image Retrieval. University of Northumbria at Newcastle, 1998.

[CWH+01] Zheng Chen, Liu Wenyin, Chunhui Hu, Mingjing Li und HongJiang Zhang. *IFind: A Web Image Search Engine.* In: Research and Development in Information Retrieval, 2001. SIGIR2001 Demo.

[DWE02] Joachim Diepstraten, Daniel Weiskopf und Thomas Ertl. *Transparency in Interactive Technical Illustrations.* Computer Graphics Forum (Proc. of Eurographics 2002), 21(3):317–325, September 2002.

[Ebe03] Davis S. Ebert. *Scientific Illustration → Part2: Illustrative Visualization*. In: Theory and Practice of Non-Photorealistic Graphics: Algorithms, Methods, and Production Systems, SIGGRAPH 2003 Course Notes, New York, 2003. ACM SIGGRAPH.

[GG01] Bruce Gooch und Amy Gooch. *Non-Photorealistic Rendering*. A K Peters, Ltd., Natick, 2001.

[GPS99] Gene Golovchinsky, Morgan N. Price und Bill N. Schilit. *From Reading to Retrieval: Freeform Ink Annotations as Queries*. In: Proceedings of the ACM-SIGIR Conference on Research and Development in Information Retrieval 1999, pp. 19–25, New York, USA, 1999. ACM Press.

[HIR$^+$03] Nick Halper, Tobias Isenberg, Felix Ritter, Bert Freudenberg, Oscar Meruvia, Stefan Schlechtweg und Thomas Strothotte. *OpenNPAR: A System for Developing, Programming, and Designing Non-Photorealistic Animation and Rendering*. In: Proc. of Pacific Graphics 2003, pp. 424–428, Los Alamitos, CA, 2003. IEEE.

[HSHS02] Knut Hartmann, Stefan Schlechtweg, Ralph Helbing und Thomas Strothotte. *Knowledge-Supported Graphical Illustration of Texts*. In: Proceedings of AVI 2002, pp. 300–307, New York, USA, 2002. ACM Press.

[JI03] Roland Jesse und Tobias Isenberg. *Use of Hybrid Rendering Styles for Presentation*. In: Proc. of WSCG 2003, pp. 57–60, 2003. Short Paper.

[LL02a] Henry Lieberman und Hugo Liu. *Adaptive Linking between Text and Photos Using Common Sense Reasoning*. In: Conejo De Bra, Brusilovsky, Hrsg., Adaptive Hypermedia and Adaptive Web-Based Systems, Second International Conference, AH 2002, Malaga, Spain, May 29-31, pp. 2–11. Springer Verlag, 2002. Lecture Notes in Computer Science 2347.

[LL02b] Hugo Liu und Henry Lieberman. *Robust Photo Retrieval Using World Semantics*. In: Proceedings of the LREC 2002 Workshop on Creating and Using Semantics for Information Retrieval and Filtering: State-of-the-art and Future Research, pp. 15–20, 2002.

[PGS98] Morgan N. Price, Gene Golovchinsky und Bill N. Schilit. *Linking by Inking: Trailblazing in a Paper-like Hypertext*. In: Proceedings of Hypertext '98, pp. 30–39, New York, USA, 1998. ACM Press.

[PHS98] Bernhard Preim, Knut Hartmann und Thomas Strothotte. *Figure Captions in Visual Interfaces*. In: Proceedings of AVI'98, pp. 235–246, New York, USA, 1998. ACM Press.

[PW97] Thomas A. Phelps und Robert Wilensky. *Multivalent Annotations*. In: European Conference on Digital Libraries, pp. 287–303, 1997.

[Scl95] Stan Sclaroff. *World Wide Web Image Search Engines*. Technical report, Computer Science Deptment, Boston University, 1995.

[SFA96] Michael J. Swain, Charles Frankel und Vassilis Athitsos. *WebSeer: An Image Search Engine for the World-WideWeb*. Technical report, University of Chicago Department of Computer Science, 1996.

[Sou03] Mario Costa Sousa. *Scientific Illustration → Part1: Traditional Techniques and NPR Approaches*. In: Theory and Practice of Non-Photorealistic Graphics: Algorithms, Methods, and Production Systems, Jahrgang 10 of SIGGRAPH 2003 Course Notes, Kapitel 2. ACM SIGGRAPH, 2003.

[SS99] Stefan Schlechtweg und Thomas Strothotte. *Illustrative Browsing: A New Method of Browsing in Long On-line Texts*. In: Proceedings of INTERACT'99, pp. 466–473. IOS Press, 1999.

[SS02] Thomas Strothotte und Stefan Schlechtweg. *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*. Morgan Kaufmann, San Francisco, 2002.

[STC97] Stan Sclaroff, Leonid Taycher und Marco La Cascia. *ImageRover: A Content-Based Image Browser for the World Wide Web*. In: Proceedings IEEE Workshop on Content-based Access of Image and Video Libraries, pp. 2–9, 1997.

[TM98] Dan Tufis und Oliver Mason. *Tagging Romanian Texts: A Case Study for QTAG, a Language Independent Probabilistic Tagger*. In: Proc. of 1st International Conference on Language Resources & Evaluation (LREC), pp. 589–596, Granada, 1998.