Adaptive Labeling for Interactive Mobile Information Systems

G. Fuchs[1], M. Luboschik[1], K. Hartmann[2], K. Ali[2], H. Schumann[1], Th. Strothotte[2]

[1]Institute of Computer Graphics
University of Rostock
D-18051 Rostock, Germany

[2]Department of Simulation and Graphics
Otto-von-Guericke University of Magdeburg
PSF 4120, D-39016 Magdeburg, Germany

Abstract

Textual annotations are important elements in all but the most simple visual interfaces. In order to integrate textual annotations smoothly into the dynamic graphical content of interactive information systems, fast yet high-quality label layout algorithms are required. With the ongoing pervasion of mobile applications these requirements are shifted from workstations to comparatively low-performance mobile devices. Fortunately, ubiquitous network access is also on the advance, so that mobile applications can employ remote layout services on external workstations. This paper presents two novel label layout algorithms for relevance-driven dynamic visualizations in interactive information systems. They are employed to generate adaptive visualizations in a mobile maintenance support scenario.

Keywords — **Mobile information systems, distributed computation platform architectures, focus & context, labeling**

1. Introduction

Due to the complexity of modern technical devices, the diversity within a product family, and frequent technological changes, providing comprehensive, up-to-date documentation becomes an increasingly challenging task. Digital maintenance manuals can supplement or even substitute their conventional (printed) counterparts. Moreover, their visualizations can be adapted dynamically to the current maintenance task at hand. This comprises (i) the application of focus & context techniques such as non-linear distortions or graphical emphasis techniques and (ii) the integration of textual annotations (labels) within dynamic content.

This combination of graphical distortions and labels is critical in that the labels must be exempted from being distorted for aesthetic and legibility reasons (cf. Fig. 1-b). Despite all advances in efficiency, the intrinsic complexity of even the simplest label layout problems still makes it a comparatively resource-intensive process. Hence, the majority of label layout algorithms in interactive applications rely on the computational power in the magnitude of typical workstations.

The ongoing pervasion of mobile devices like PDAs and smartphones makes it increasingly expedient to offer applications such as digital manuals as part of mobile information systems. Compared to stationary computers, mobile devices are still seriously limited in both computational power and storage capacity. Especially when running full-fledged applications on such devices, executing complex labeling algorithms is very likely to pose a serious performance bottleneck. Fortunately, the bandwidth and availability of wireless network access is growing at a fast pace. Thus, it becomes feasible to make use of distributed computation platform (DCP) architectures, where complex tasks are shifted to external servers and the client integrates their results.

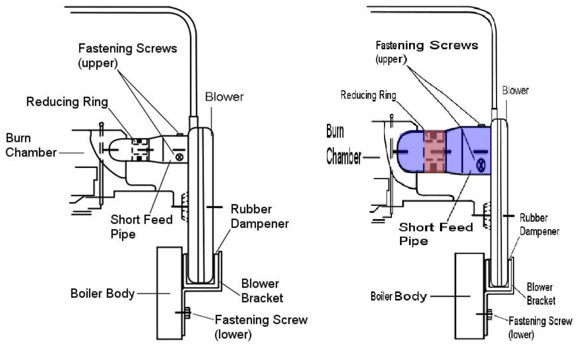
Therefore, this paper is focused on an adaptive selection and layout of labels in interactive mobile information systems. Both the use of a remote labeling server, and a resource-conservative on-device approach are considered.

This paper is structured as follows: First, we analyze the requirements and challenges of adaptive visualizations and dynamic label layouts on mobile devices (Sec. 2). Section 3 proposes new solutions and presents a distributed computation platform architecture to integrate label layouts on visualizations on mobile devices and several space-efficient label layout algorithms (Sec. 4). Section 5 summarizes this paper and motivates directions for future research.

2. Concepts for Task-Based Visualization in Mobile Maintenance Applications

Our approach extends task-based visualizations in a mobile information system [6]. We describe the application scenario and give a brief overview of its implementation. Subsequently, we present the advantages of an automated layout of textual annotations for adaptive visualizations as used in this system.

Proceedings of the Information Visualization (IV'06)
0-7695-2602-0/06 $20.00 © 2006 **IEEE**



(a) Original (b) Distorted

Figure 1. An illustration with static labels (left). Distorted labels due to focus & context techniques (right).

2.1. Scenario

We have chosen maintenance operations on air-condition units as a promising case study, as these devices have comparatively long life cycles and often receive mid-life upgrades to meet current energy efficiency standards. Mobile 'e-manuals' assist technicians to gain sufficient knowledge of all peculiarities of different units: modified schematics, operational procedures, and assembly instructions. Dynamically generated illustrations adapted to a given maintenance task on hand support the technician; printed (paper) manuals would be far too bulky for this purpose.

Task models represent operational and maintenance procedures in semantic networks. These models determine the relevance of components of the air-condition unit within the steps of a maintenance operation. Adaptive visualizations apply graphical emphasis techniques to highlight modifications on affected components or the instruments needed within each step of a task sequence. Moreover, the user interface in general can be adapted to that task (dialog transitions, input masks). Instructions and descriptions are provided by speech output and in a separate text display. Depending on the hardware capabilities, speech input may be also available. A more detailed description of the project setting and a prototypical implementation is provided in [6].

Adaptive visualizations. According to Shneiderman's well-know mantra [12], a visualization should provide both an overview and details on demand. The need of an interactive exploration and filtering by the user is greatly reduced by the backing task model, since the definition of task goals largely removes the necessity for (undirected) searches in information space. However, due to the small screen sizes, the information density of a given presentation has to be carefully managed for the visualization to remain effective.

Non-linear distortion techniques enable comprehensive detail-in-overview representations of large images on small screens. However, the performance bottleneck faced on almost all current mobile devices discourages the use of arbitrary, potentially complex transformation functions. Therefore, a belt-based focus & context technique for bitmap images was used in the current prototype [10, 8]. Instead of continuous transformations, it uses rectangular regions (belts) of constant magnification (cf. Fig. 1-b). It exhibits adequate results with an acceptable performance impact.

The majority of mechanical drawings constituting the graphical content of a technical manual employ only two intensity values: black and white. Therefore, graphical emphasize techniques can also modify the opacity, saturation, or color hue in order to guide the viewer's attention to salient regions.

2.2. Labeling in Multi-Modal Interfaces

The interface of mobile information systems primarily relies on visual representations. However, highly domain-specific concepts or technical terms may require explicit textual annotations in an image. Therefore, labels are an important element of effective technical illustrations. Furthermore, labels serve two additional purposes in multi-modal interfaces:

a) Coordination of the visual and audio modes. Textual annotations in images link elements of the visual representation with information conveyed via speech, e.g., by providing corresponding catchwords or captions. If speech input is available, the presentation of keywords in labels establish an intuitive access/control mechanism for voice commands [3].

b) Redundant information on the visual and auditive channel. Labels convey additional information in situations when the ambient noise level at the maintenance site overpowers the speakers. Moreover, labels may remain on the interface permanently without introducing a comparative "auditive clutter" by constant repetitions.

Thus, labels are an integral part of effective, adequate visualizations in multi-modal interfaces. However, these labels must be placed dynamically, i.e. at runtime. Static labels, i.e. text as integral, fixed image content, would be subjected to the distortions introduced by focus & context techniques, which is unsatisfactory both from an aesthetic point of view and for legibility issues (cf. Fig. 1-b). Dynamic labeling, as opposed to static labels, therefore has to be adaptable in two ways:

a) Adaptive with respect to content: The number of labels, their size, position and, of course, the displayed text, are dependent on the information encoded in the illustration. An effective layout has to exploit unused background areas while preventing label overlaps. Finally, the label's visual attributes may also communicate the relevance of their associated visual objects.

b) Adaptive with respect to technical restrictions: Label layout algorithms for interactive systems have to consider the trade-off between quality and response time. Small imperfections are tolerable when the content, and thus the labels, are in return modified at interactive rates. Moreover, resources must be shared between different processes, and the label layout likely does not have the highest priority. Therefore, the number, size, and positional accuracy of placed labels may have to be reduced on low-power clients, or with only low network bandwidth available (in client/server scenarios).

3. Adaptive Label Layout Architectures

We implemented two strategies in order to perform adaptive labeling under the considerations set forth in the previous section:

- Create an adequate overall labeling of the entire image, or
- label only certain details on demand, by means of a *labeling lens* interaction tool.

Adaptive label layouts. The first approach determines a label layout according to the current task at hand. Although it potentially provides a good overview (all relevant objects are labeled simultaneously), the optimization task might be too computationally complex for mobile devices. Aside from having to render the illustrations, the mobile device is also tasked with running the speech server and the task model/host application. Therefore, we make use of a client/server architecture to process the adaptive label layout with respect to several metrics externally.

Moreover, to be able to support such remote labeling for arbitrary visualizations, the concept of layers was adapted. Here, the client creates the distorted, adapted illustration that does not contain any labels. In parallel, all information necessary to create appropriate annotations are transmitted to the server. A description of the position, size, and text of all labels placed is returned. This information is sufficient for the client to render an *annotation layer* that is superimposed over the visualization.

The **labeling lens** approach reduces the number of objects to be labeled and thus the complexity of the layout problem. Labels are only added for features in the confined region of the lens tool. The user can interactively move the lens to reveal labels in currently unlabeled regions of the image. This strategy can either replace or substitute a(n)(unavailable) label server.

Before presenting a detailed description of the layout algorithms in the next section, we describe the process flow in the distributed computation platform architecture approach.

Process flow. Fig. 2 shows the integration of dynamic labeling into the mobile maintenance support scenario. Scanned manuals are the input source for the majority of technical drawings. In order to allow dynamic annotations, labels and connecting lines are removed from the illustration (cf. Fig. 3-a).

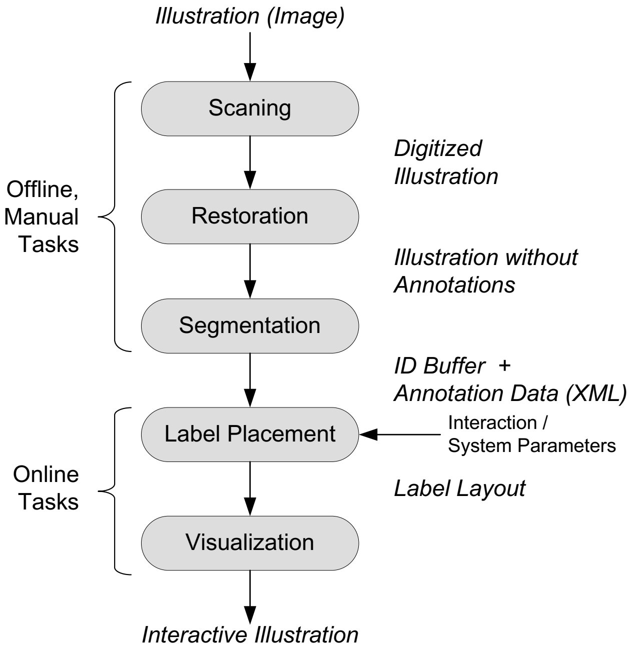


Figure 2. Process flow.

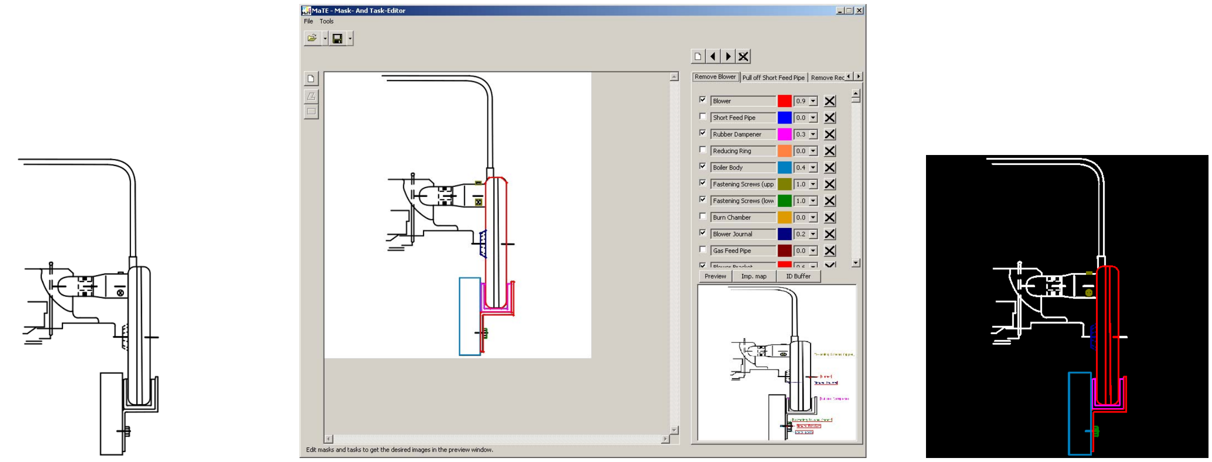
Subsequently, the images are annotated by marking specific regions as features to be labeled later. Thus, the main effort of annotating is the segmentation of images into regions with semantically meaningful content. Even though there exits a number of automatic or interactive approaches, the image segmentation remains a very complicated task.[1] As our main research focus lies on developing adaptive user interfaces and not on image-processing techniques, we developed two interactive tools. One enables authors to create new task models and to specify tasks and associated work items. The second tool is used to mark spatial regions of images as features that can be associated with tasks or work items (cf. [6]). Here, users can assign a label text to each feature and define their relevance for specific tasks (cf. Fig. 3-b). These information are stored in XML structures.

The label layout algorithms purely rely on the image segmentation and the association of colors with label texts. Therefore, ID buffers are generated dynamically, where distinct objects are identified by unique colors (see Fig. 3-c), as they have to reflect the current status of the visualization (e.g., non-linear distortions, dynamic changes to the set of

---

[1] The efficient segmentation of images comprises standard techniques (e.g., threshold techniques, edge-based methods, region-based techniques, connectivity-preserving relaxation methods) as well as sophisticated model-based or graph-based optimization techniques. The limited space prevent an in-depth review or comparison.

0-7695-2602-0/06 $20.00 © 2006 IEEE



☑	Blower	🟥	0.9
☐	Short Feed Pipe	🟦	0.0
☑	Rubber Dampener	🟪	0.3
☐	Reducing Ring	🟧	0.0
☑	Boiler Body	🟦	0.4
☑	Fastening Screws (upp	🟩	1.0
☑	Fastening Screws (low	🟩	1.0
☐	Burn Chamber	🟧	0.0
☑	Blower Journal	🟦	0.2
☐	Gas Feed Pipe	🟥	0.0

Edit masks and tasks to get the desired images in the preview window.

(a) Restored illustration without annotations.

(b) Mask and Task Editor.

(c) Distorted ID-Buffer.

Figure 3. Manual image restoration and segmentation.

'active' features). Subsequently, both the distorted ID buffer and the XML annotation data are sent to the label server. After user interactions, either a procedural description of change to the visual content, or an updated ID buffer have to be sent to the server. Moreover, both the relevance values as well as the label content may be updated. The label server returns a XML specification of the label layout to the client. It contains label positions and the coordinates of anchor points and connecting lines. The PDA client reads this XML stream and superimposes its rendition over the visualization in interactive time. The following sections will discuss the label algorithms in detail.

4. Label Layout Algorithms

This section first provides definitions for some terms used informally so far and discusses functional and aesthetic requirements posed on an effective label layout. The next subsections then sketch the implementation of two strategies to integrate an adaptive labeling into the dynamic content of mobile information systems: the determination of a label layout for the entire illustration and a label layout for those objects in the focus of an interactive tool.

4.1. Terminology and Requirements

Labels may either overlay their reference objects or are placed outside (*internal* vs. *external* labels). Moreover, additional *secondary elements* may highlight the co-referential relation between a textual label and its associated *primary* visual elements of the object.

Area or *line features* can accommodate internal labels. This label class does not need any secondary elements to establish the co-referential relation. The text stroke can highlight the shape of its visual reference object, especially, if there are no distinguishable boundaries between individual objects. However, internal labels may hide important visual features and are hard to read if text strokes are curved, not aligned to the main axis, or the text has an insufficient contrast to the background.

External labels are placed on the *empty space* which is not covered by primary elements (background). Furthermore, their text strokes are aligned to the main axis of the illustration. While internal labels tightly integrate textual and visual elements, external labels separate elements of both media from each other. Therefore, *connecting lines* have to link depictions and textual descriptions, while *anchor points* ease the identification of the visual reference object. In many illustrations, however, primary and secondary elements share the same visual code – lines.

Tufte's *most effective difference principle* [13] advises illustrators to reduce the visual clutter introduced by secondary and *structural* elements. An optimal layout should ease the figure-ground differentiation, provide clear distinctions between primary and secondary elements, minimize the difference between secondary and (back)ground elements, and structure the layout hierarchically. This could be achieved by a uniform alignment, spacing, or coloring of secondary elements. Probably, that is the reason why scientific and technical illustration almost exclusively employ external labels while internal labels are used pervasively in cartography.

An effective and aesthetic label layout must meet various constraints: the layout must guarantee that the viewer can extract the correct co-referential relations between graphical and textual elements (*unambiguity*), the layout elements should ease the *readability*, and in interactive application, the computation has to be *efficient*. Moreover, the distance between layout elements in subsequent frames should be minimized (*coherency*). In mobile applications, the small

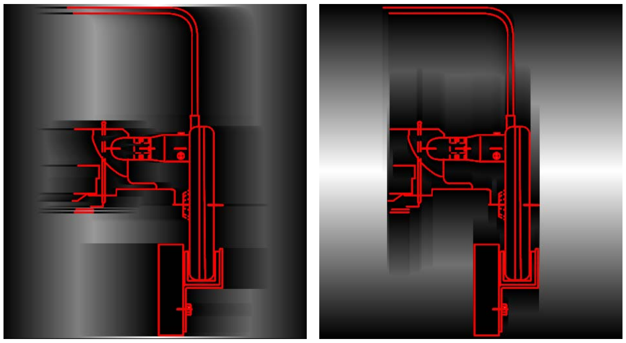


Figure 4. Distance fields in x- (left) and y-direction (right).

screen size poses an additional requirement — the layout must be *space-efficient* to fit within the available space. Note that these requirements might conflict with one another. Therefore, a layout is sought which balances these requirements. These label layout algorithms heavily rely on heuristics, as an even simpler problem (finding an optimal layout for the point-feature labeling problem) has been proven to be NP-hard [9].

There exists a number of label layout algorithms designed for interactive applications (e.g., [5, 1, 4] for external or [7] for internal labels and Bell's efficient space-management algorithm [2]), however, they did not achieve sufficient results on the restricted display size of mobile devices. The majority of these algorithms employ approximations of the object's shape (either bounding boxes [2] or convex hulls [1, 4]) in order to determine and manage empty space. In most circumstances, the empty space on small visualizations forms disjoint isolated regions, which are also covered by the space approximation. This forced us to develop two novel space-efficient layout algorithms, which either modify distance functions or refine Bell's space management algorithm.

4.2. Asymmetric Distance Fields

The fast and accurate determination of the empty space between primary elements in an illustration is essential for any external label layout algorithm. Hence, our new space management algorithm determines empty regions in the image spaces, which is more accurate than any algorithm based on bounding objects (object space).

Using the ID-buffer we first perform a fast *distance transformation* to propagate the distance to the objects' silhouettes for all pixels in linear time. We use a fast distance transformation [11] which incorporates a Euclidean distance metric and results in a *distance field*. We modify the standard distance transformation algorithms in order to get the minimal distance in x and y direction (cf. Fig. 4-left and Fig. 4-right). The label placement is done in a greedy manner: we select the most important label according to the task model, determine maximal rectangles of empty space, place the selected label in the nearest region and update the distance fields. The results from our system (cf. Fig.5-left) could be compared to the layout done using *Left and Right* layout style proposed by [1]). The previous approach (cf. Fig. 5-right) relied on first placing the labeling strictly on the left and right sides of the graphical object and then bringing them close to the silhouette boundaries without taking into account the space that is already available on the image. Our new approach explicitly looks for the nearest locations to the objects where the corresponding labels could be placed and thus makes efficient use of existing space without increasing the illustration size.

Using this algorithm the empty space is used very efficiently even if it is separated into small and irregular regions. To improve the efficiency, the expensive update of the distance function after a label placement was substituted by the efficient empty space management of Bell's original algorithm [2]. Unfortunately, we cannot avoid label/line intersections and the irregular distribution of labels may reduce the aesthetic quality of the layout.

4.3. Labeling Lens

In order to implement a real-time label layout algorithm on the restricted resources of a mobile device that has to share both processing and storage capacities with other processes, we had to reduce the complexity of the search space even further. Therefore, we target to label only objects selected by an interactive tool – the *labeling lens*, a small movable selection tool where labels are placed without obscuring any visual object. This tool was inspired by the excentric labeling approach [5], a powerful visualization for large data sets, where only the direct neighbors of an interactively selected object where labeled on a circular orbit. Fekete and Plaisant argued that this interactive labeling interface increases the working efficiency, but their approach does not consider the relevance of data occluded by labels. This labeling style in an 'e-manual' would prevent the integration of detailed into contextual information – it is very likely that important reference objects would be occluded by labels.

Our new approach combines and extends both the excentric labeling and the space management techniques: A complex lens consisting of two centrically aligned shapes (preferably circles or squares, cf. Fig. 6) that can be positioned above displayed details of free choice. The inner shape is a classical selection lens to define the objects which are to be labeled. The outer shape is used to restrict the space of label placement. To achieve a conflict-free labeling inside this lens, the empty space inside the lens area is

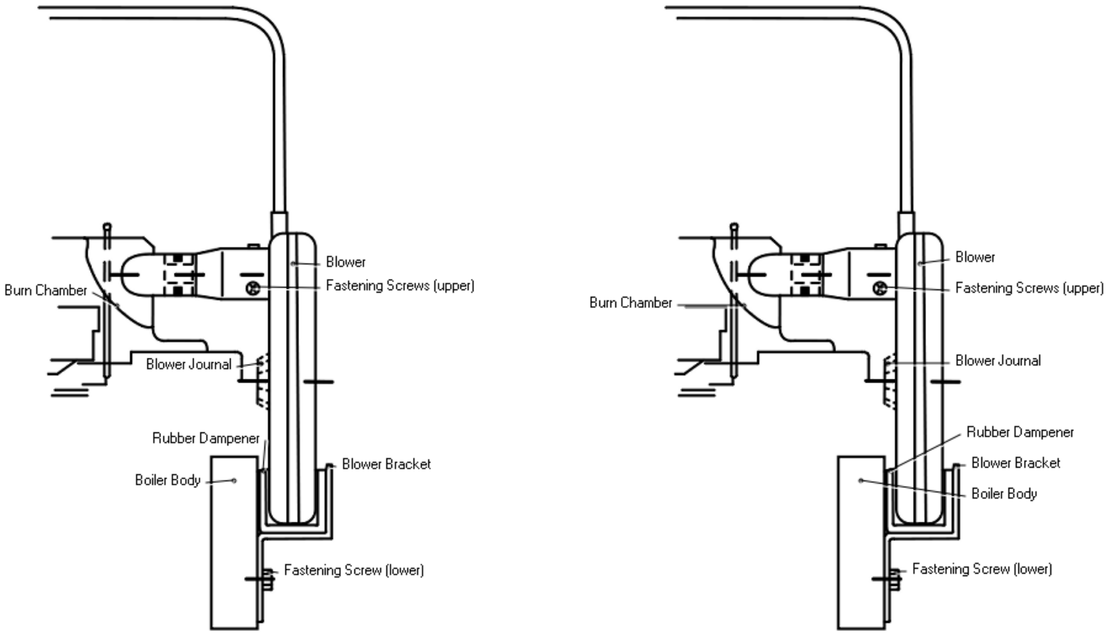


Figure 5. Adaptive labeling using empty space management (left) and a *Left and Right* style (right).

determined with the scan-line algorithm. The labels of selected details are placed within the smallest fitting empty rectangle closest to the selected feature and connected to them with connecting lines (cf. Fig. 7-c).

We modify Bell's space-management algorithm [2] which approximates the space used to display primary objects with axis-aligned bounding boxes. The *precision* of this approximation, i.e. ratio between the exact object area and the area of the bounding object, can be quite low for complex-shaped objects. Thus, in many situations, the remaining empty space is separated into very narrow or small rectangles. This often leads to unbalanced layouts where the distance between the labels and there reference objects is quite high (cf. Fig. 7-a).

A decomposition of complex-shaped objects into simple components can improve the precision of the bounding object. Moreover, we loosen the requirements from bounding objects to *shape approximations*, i.e. representations that may not cover the entire area of visual objects. Therefore, we sample the image space on equidistant scan-lines and determine the spans of used spaces. The used space is approximated by rectangles; their heights is equivalent to the distance between scan-lines. The remaining empty spaces are than managed with Bell's original algorithm (cf. Fig. 7-b). The distance between scan-lines can be adapted to minimize the computing time or to maximize the precision.

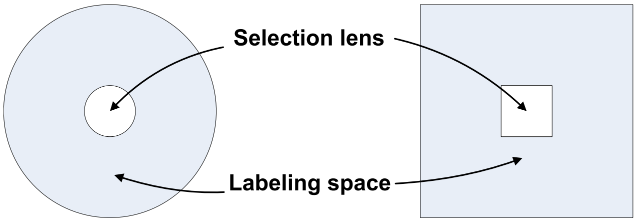


Figure 6. Labeling lenses in different shapes.

Thus, the labeling quality and computational complexity can be adjusted according to the distance between scan-lines, the diameter of the lens, and the lens type. Here, the type is either a simple selection lens as for excentric labeling, or the conflict-free labeling lens described above. By this, the labeling quality can be dynamically adapted to the available resources on a mobile device.

5. Conclusion and Future Work

This paper presents two novel label layout algorithms for adaptive information systems. Both layout algorithms consider the restricted resources on mobile devices (i.e. the display sizes, processing and storage capacities). Specifically, the visualizations are adapted to support specific maintenance tasks. Therefore, the relevance of the visual objects and the content to be displayed on textual labels are inferred from a formal task model. Both algorithms are incorporated in a distributed computation platform architecture and are used in an industrial application prototype. The software has been presented at the CeBIT fair and received positive responses.

So far, the label layout facilitates only external labels. We excluded internal labels as their text strokes has to overlay the reference object. Hence, their layout has to consider the object's shape in image space, and their position has to be

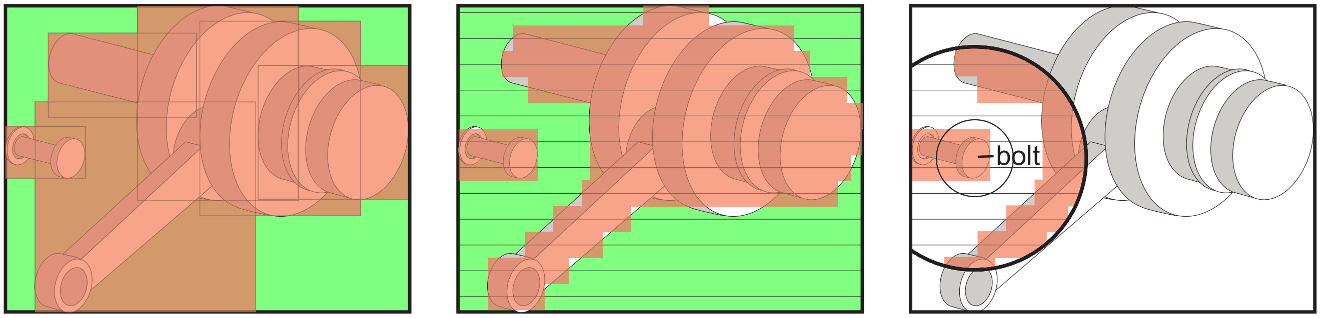


Figure 7. ***Left:*** **Empty (green) and used (brown) space according to Bell's space management algorithm [2] for six visual objects.** ***Center:*** **Our scan-line approach employs much smaller bounding rectangles and thus reduces prodigality of empty space.** ***Right:*** **A labeling lens.**

updated immediately after user interactions. However, internal labels are a valid alternative if there is not enough empty space to place external labels. Therefore, we plan to exploit the scan-line shape approximation also for the placement of internal labels, so that they can be directly computed on the mobile client.

Even though the current implementation can use bitmaps and Scalable Vector Graphics (SVG), the majority of technical drawing still remains in a bitmap format, owing to the difficulties in vectorizing scanned images. However, using SVG would allow to inline the necessary XML annotations directly into the image file, and to define image features by referencing groups of vector primitives. Also, the dynamic renditions of vector graphics would allow more sophisticated distortion functions. For these reasons we plan to create another tool to edit/enrich SVG files accordingly.

Moreover, the label layout server is unable to create the focus & context distortions itself, so that the mobile client has to send distorted color-code bitmaps to the label server after each corresponding interaction. Therefore, we currently port the belt-based non-linear distortion from the client to the server, in order to minimize the communication overhead.

References

[1] K. Ali, K. Hartmann, and T. Strothotte. Label Layout for Interactive 3D Illustrations. *Journal of the WSCG*, 13:1–8, 2005.

[2] B. Bell and S. Feiner. Dynamic Space Management for User Interfaces. In *Symp. on User Interface Software and Technology*, pages 238–248, 2000.

[3] N. Biehl, A. Düsterhöft, P. Forbrig, G. Fuchs, D. Reichart, and H. Schumann. Advanced Multi-Modal User Interfaces for Mobile Devices — Integration of Visualization, Speech Interaction and Task Modeling. In *Information Resources Managment Associciation Int. Conf. (IRMA)*, 2006. (short paper).

[4] S. Bruckner and E. Gröller. VolumeShop: An Interactive System for Direct Volume Illustrations. In *IEEE Visualization*, pages 671–678, 2005.

[5] J.-D. Fekete and C. Plaisant. Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization. In *SIGCHI Conf. on Human Factors in Computing Systems*, pages 512–519, 1999.

[6] G. Fuchs, D. Reichart, H. Schumann, and P. Forbrig. Maintenance Support — Case Study for a Multimodal Mobile User Interface. In *IS&T/SPIE's 16th Ann. Symp. Electronic Imaging: Multimedia on Mobile Devices II*, 2006.

[7] T. Götzelmann, K. Ali, K. Hartmann, and T. Strothotte. Form Follows Function: Aesthetic Interactive Labels. In *Computational Aesthetics 2005. EG WS on Computational Aesthetics in Graphics, Visualization and Imaging*, pages 193–200, 2005.

[8] B. Karstens, R. Rosenbaum, and H. Schumann. Information Presentation on Mobile Handhelds. In *Information Resources Managment Associciation Int. Conf. (IRMA)*, 2003.

[9] J. Marks and S. Shieber. The Computational Complexity of Cartographic Label Placement. Technical Report TR-05-91, Center for Research in Computing Technology, Harvard University, 1991.

[10] U. Rauschenbach, S. Jeschke, and H. Schumann. General Rectangular Fisheye Views for 2D Graphics. *Computers & Graphics*, 25(4):609–617, 2001.

[11] F. Y. Shih and Y. Wu. Fast Euclidean Distance Transformation in Two Scans Using a 3x3 Neighborhood. *Computer Vision and Image Understanding*, 93(2):195–205, 2004.

[12] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization. In *IEEE Symp. on Visual Languages*, pages 336–343, 1996.

[13] E. R. Tufte. *Visual Explanations: Images and Quantitatives, Evidence and Narrative*. Graphics Press, Cheshire, Connecticut, 1997.