Eurographics Workshop on Visual Computing for Biology and Medicine (2012)
T. Ropinski, A. Ynnerman, C. Botha, and J. Roerdink (Editors)

Constrained Labeling of 2D Slice Data for Reading Images in Radiology

Katja Mogalle[1,2], Christian Tietjen[2†], Grzegorz Soza[2], and Bernhard Preim[1]

[1]Department of Simulation and Graphics, University of Magdeburg, Germany
[2]Siemens Healthcare Sector, Forchheim, Germany

Abstract

An important and underestimated task to support reading of images in radiology is a proper annotation of findings. In radiology reading, 2D slice images from a given modality (e.g. CT or MRI) need to be analyzed carefully by a radiologist, whereas all clinical relevant findings have to be annotated in the images. This includes information in particular for documentation, follow-up investigations and medical team meetings. The main problem of the automatic placement of labels in a clinical context is to find an arrangement of multiple variable-sized labels which guarantees readability, clearness and unambiguity and avoids occlusion of the image itself. Based on a case study of abdominal CT-Images in an oncologic context we analyze the main constraints for label placement in order to extract candidate label positions, evaluate these and determine valid and good label positions. Based on this preprocessing step, different approaches can be applied for placing multiple labels in a scene. We present a new method called Shifting *and compare it to other labeling strategies.*

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

1. Introduction

In radiology, images that have been acquired from any modality, need to be read carefully by the radiologist. Irrespective of the reason for the particular image acquisition, the radiologist has to document all potentially pathologic findings. Because reporting findings solely in the form of a text document can be misleading, it is also recommended to report all findings directly in the image data, more precisely onto the 2D slice. To do so, findings are illustrated with according graphics (segmentation contour, distance line, etc.) and a label depicting the measurement values needed for documentation (volume, diameter, etc.).

The main goals of annotations are to maintain a clear representation of findings and simplify the process of reviewing clinical images during follow-up measurements, consultation of experts as second opinion and patient education. Thereby different representations of labels implicate different challenges concerning layout and screen space which have to be solved during the labeling process. The manual placing of labels by the radiologists is time-consuming and may have to be readjusted in every frame displaying a different composition of the scene. Simple algorithms for placing labels, however, face the problem of invalid or bad arrangements and lead to great distraction during navigation. Especially in radiology reading it is important that the pathologic findings are not occluded by graphical objects in order to analyze the image data precicely. We therefore present an algorithm for placing labels in 2D slice data taking into account readability, unambiguity, clarity and a fast computation in order to support radiologists at diagnostics. Moreover, we consider coherency in labeling positions between adjacent slices. Most labeling techniques developed for medical image data focus on 3D models, e.g. for anatomy, education or surgical planning [MP09, RPRH07, AHS05]. We are aiming at an approach that can handle a realistic setting of reading an oncologic case. In those cases, up to ten lesions have to be measured, with up to five lesions in one organ [TAE*00]). The latter aspect causes many labels on little space. To handle those cases correctly, our algorithm is able to find an optimal position for five to ten labels in a few milliseconds.

This paper is organized as follows: after discussing related

† christian.tietjen@siemens.com

work in Section 2 we introduce special terms relevant for labeling (Sect. 3) and describe the constraints in the labeling task (Sect. 4). Based on these requirements we present the parts of the algorithm for arranging labels in Section 5. A new strategy for placing multiple labels is thereby introduced. After evaluating the presented approaches in Section 6 we show the integration of the best algorithms in a labeling system which is adapted to the clinical workflow in oncology (Sect. 7). Finally the conclusion of our work is presented in Section 8 as well as a short outlook.

2. Related Work

The labeling task we address is related to the well-studied map labeling problem which states a NP-hard computational complexity [MS91]. Christensen et al. [CMS95] summarize multiple strategies (exhaustive search, greedy approaches, gradient descent methods, nonlinear optimization, stochastic search) which can be used to label point-features and fulfill the trade-off between the quality of the label arrangement and computation time.

These strategies can be adapted to labeling illustrations and computer-generated models or images. Ali et al. [AHS05] analyzed handmade illustrations which showed that the mainly used label types are internal and external labels. The textual element of internal labels is drawn directly on the ROI whereas external labels are placed outside the annotated object and use connection lines to denote the correspondence. Internal labels or a mix of both are often applied in complex virtual 3D scenes [MD08], augmented reality applications [BFH01] and especially in the medical field [RPRH07] for educational purposes (e.g. textbooks). The task of placing external labels in simple scenes with compact models is quite different to point-feature labeling in maps. This results from the abolishment of the restriction that each label can only be placed at a small number of positions directly adjacent to the point-feature. The new degrees of freedom implicate a large amount of restrictions to the label arrangement to guarantee readability (no overlapping graphical elements), unambiguity (easy matching between label and its associated object), aesthetic layout (no visual clutter), real-time ability (computation at interactive rates) and frame-coherency (no flickering of labels on successive frames). Ali et al. present simple solutions for labeling interactive 3D illustrations with respect to different layouts. The compact models are thereby embedded in large image plains with much space for placing multiple single-line labels. Besides placing labels by a constructive algorithm another method for creating pleasant label layouts is to formulate all criteria mathematically and solve the resulting optimization problem greedily [ČB10]. The limitations hereby are that all label positions have to lie on a specific hull geometry which is achieved by the use of compact models in an unbounded image plane.

In contrast to that, Fuchs et al. [FLH*06] investigated the problem of label placement in space and time critical applications such as mobile devices. The computational effort can be reduced by using a greedy algorithm for placing labels successively in the nearest maximal rectangles of empty space [BF00]. Another approach which uses the free space in between the illustration in order to avoid the expansion of the image border is called floating labels [HAS04] which uses different forces to place labels by diffusion.

In most applications, complex scenes composed of many objects have to be labeled which also allow user interaction for exploration. The main challenge is to reduce occlusion due to intersections between graphical objects (line-line, line-label, label-label and label-scene) and above all dealing with differently-sized labels. Stein and Décoret [SD08] try to address all these issues by modeling a minimization problem (containing recommendations for a good placement) under constraints (hard constraints to achieve a valid arrangement).

3. Labeling of Medical Image Data

The typical tools which support the viewing, reading and reporting of 3D medical images include four viewports showing three orthogonal 2D slices at a user-specified position and a 3D rendering. We only focus on placing labels in 2D images, e.g. retrieved by multiplanar reconstruction in CT imaging, because of the high relevance in clinical applications. In Figure 1 the main terms are illustrated which are necessary for describing the labeling task. An annotation comprises all graphical elements used to accentuate and label a ROI. This especially includes a sample point of the ROI called *anchor*, the label and the graphical connection between label and object. A label is defined by its textual content embedded in the label box which is specified by the label size and label position (center of box).

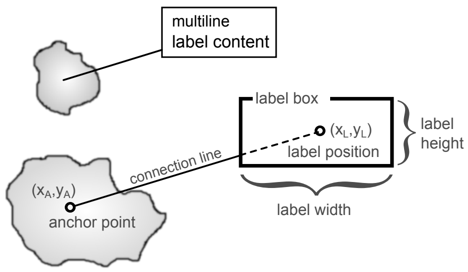


Figure 1: *Illustration of important terms for the task of automatic label placement.*

A viewport is shown in Figure 2 containing the image text at the viewport border (scale, orientation letters, patient data, etc.) as well as the 2D slice with visually emphasized findings and their labels. The user is able to select any view and position in the dataset by zooming, panning, rotating and changing the current slice via scrolling (slicing). Thereby a label has to be placed for every finding visible in the current view. We prefer the use of external labels drawn directly on the slice. These labels are not placed on top of the corresponding ROIs but rather outside the image.

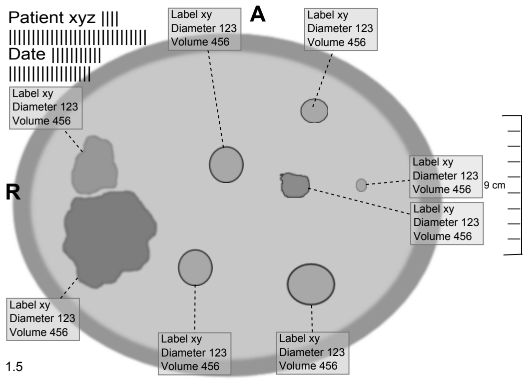


Figure 2: *A 2D slice with image, elements of the image text, emphasized ROIs and corresponding labels.*

4. Requirements Analysis

We investigate the integration of external labels in the reading and reporting process of medical cases during diagnostics. Therefore different requirements have to be met which result from several discussions with radiologists, experiences with their current software and an analysis of textbook illustrations as well as related work. The constraints are categorized in general (G) requirements, requirements for placing single (S) or multiple (M) labels and requirements concerning the behavior (B) of labels during user interaction.

Number of labels (G1) - In a typical oncologic 3D dataset we expect a maximum number of ten marked objects in a single 2D slice due to the RECIST criteria [TAE*00].

Fast computations (G2) - The interactivity of the diagnosis software may not be affected by the computation of label positions and visualization of the visible labels.

Border placement (S1) - All visible labels should be placed as close as possible to the viewport border in order to ensure a free sight on the image part focused at the viewport center and to minimize the distraction by the labels overlying the image during the reading process.

Avoid overlap of image texts (S2) - The automatically placed labels may not intersect image texts which are displayed on top of the image (e.g. patient data, orientation letters and scales) to guarantee readability.

Avoid overlap of image parts (S3) - The automatically placed labels should also not overlap important parts of the image. This primarily includes the occlusion of any findings (**S3(i)**). Furthermore labels should not overlap any other parts the acquired image and be preferably placed on the black background (**S3(ii)**).

Line orientation (S4) - Connection lines should be aligned parallel to the viewport borders (horizontally or vertically) for a clear arrangement and to reduce ambiguity.

Line length (S5) - The distance between a label and its corresponding object (anchor point) should be minimized to reduce ambiguity and the cognitive effort for identifying label-object pairs.

Avoid overlapping Labels (S6) - Automatically placed labels may not overlap any locked labels for readability.

Avoid overlapping Labels (M1) - Automatically placed labels may not overlap any unlocked labels for readability.

Avoid intersecting lines (M2) - The straight connection lines may not intersect each other to reduce ambiguity.

Adding annotations (B1) - During reading, newly added annotations have highest priority due to the momentarily increased focus and the label should thereby receive the best possible position.

Locked Labels (B2) - Users should be able to place labels manually which denotes these labels as locked. Locked labels are not placed automatically and may not be covered by other automatically (unlocked) placed labels.

Navigation (B3) - During navigation findings may appear, disappear or move in the viewport whereby automatically placed labels may often change position in sequent frames. Distraction due to flickering should be suppressed.

Due to the numerous (S)- and (M)-requirements we classify these in **mandatory** requirements A={S2, S3(i), S6, M1, M2} and **optional** requirements B={S1, S3(ii), S4, S5}.

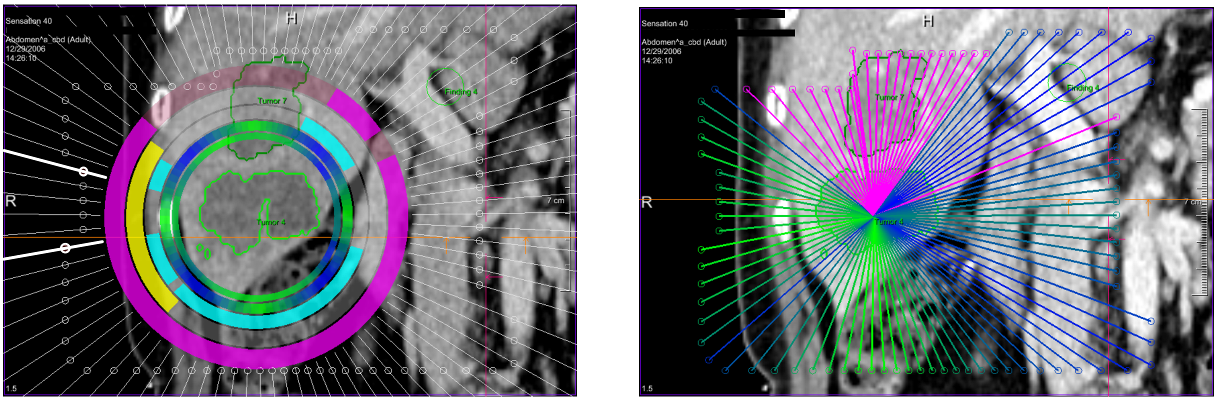
5. Methods

The method for automatically placing labels is composed of three steps. In the first step candidate positions have to be computed for each visible label. All these positions are evaluated in the second step concerning the (S)-requirements so that an optimal position can be chosen for single labels. The main labeling task is examined in the third step where one candidate position is selected for each label with the goal of placing multiple labels with respect to the (M)-requirements.

5.1. Candidate Label Positions

The placement of many labels is very complex since the search space for each label position comprises the complete 2D space bounded by the viewport border. The complexity of the problem can be reduced by restricting the search space in two steps.

In the first step we only allow positions which fulfill the requirements S1 and S2. The search space is reduced by S1 to positions on a rectangle given by the downscaled viewport border. All labels on this rectangle touch the viewport



(a) The rings around the highlighted tumor illustrate the compliance of the requirements S3(i), S3(ii), S1, S4 and S5 from outside to inside. The colored parts of the outer three rings denote the complete fulfillment of the specific requirement whereas the green parts of the two inner rings depict a high and the blue parts a low compliance.

(b) This visualization shows the result of the combination of all (S)-requirements as colored rays. The pink rays represent invalid candidate positions with $E(r) = -1$. The valid positions are color-coded from blue (worst positions with $E(r) = 0$) to green (best positions with $E(r) = 1$).

Figure 3: *Visualization of candidate positions (circles), the compliance of (S)-requirements and the resulting quality values for candidate positions for the label of a specific finding.*

border. Due to different label sizes this rectangle has to be defined for each label separately. Many of these positions violate S2 ans thus we adapt the indented rectangles to meet S2 by integrating parts of enlarged bounding boxes of the single image texts. This leads to a contour called *adaptive border* (see dotted polyline in Fig. 4) which can be computed for each label and contains only label positions where the specific label does not overlap any image texts and is as close to the viewport border as possible.

In the second step we select a subset of the positions on the adaptive border because the screen resolution is too high for an efficient computation. We therefore sub-sample the adaptive border by sending rays from the viewport center to the border and computing the nearest intersections with the adaptive border (see Fig. 3). The number of rays can be changed by the user to adjust the trade-off between computational complexity and accuracy or quality. The default value we used for all examinations was $R = 90$ rays.

Due to the restriction of the search space to a finite number of candidates on the adaptive border, the requirement S2 becomes superfluous. Moreover, S1 can be restated such that a label should preferably touch the viewport border.

5.2. Placing Single Labels

We first consider the optimal position of a single label without taking into account any other labels. Therefore the candidate positions only have to be evaluated with regard to the (S)-requirements. Thus six functions $f_j(r)$ with $j \in \{S1, S3(i), S3(ii), S4, S5, S6\}$ are defined which convey the degree of compliance for each (S)-requirement. For a particular label and a candidate position defined by its ray index r with $0 \leq r \leq R$, the function $f_j(r)$ with $0 \leq f_j(r) \leq 1$ expresses if the requirement j is best met ($f_j(r) = 0$) or is most poorly fulfilled ($f_j(r) = 1$) with respect to the other candidate positions. These relationships are visualized in Figure 3(a). Based on this definition, a single function $g(r)$ can be specified which merges the compliance of all six requirements into one measure:

$$g(r) = \begin{cases} -1 & \text{, if a mandatory requirement (S3(ii) or S6) is not fulfilled} \\ \sum_{j \in B} \gamma_j * f_j(r) & \text{, else} \end{cases} \tag{1}$$

The importance of each optional requirement can thereby be manipulated by the user by adjusting the weights γ_j.

For a better understanding, the function $g(r)$ is flipped and normalized to form the objective function $E(r)$, which expresses the quality of a candidate position:

$$E(r) = \begin{cases} -1 & \text{, if } g(r) < 0 \\ \dfrac{max_g - g(r)}{max_g - min_g} & \text{, if } max_g \neq min_g \\ 1 & \text{, if } max_g = min_g \end{cases} \tag{2}$$

While $E(r) = -1$ denotes an invalid position, $E(r) = 0$ denotes the worst and $E(r) = 1$ the best rated position for the current label (see Fig. 3(b)). An optimal candidate position can then easily be obtained for a single unlocked label by selecting a ray r with $E(r) = 1$.

5.3. Placing Multiple Labels

Placing single labels is just a small aspect within labeling a whole image. Given a 2D slice, a number of N visible and unlocked labels have to be placed such that the quality values of each label position is at a maximum and requirements M1 and M2 are considered. In total there are R^N arrangements of the N labels with each R candidate positions which have to be examined. A label arrangement is called *valid* if neither labels overlap nor connection lines intersect each other. An exhaustive search testing all arrangements until a valid high quality constellation is found is too time-consuming. Hence a best possible solution has to be approximated. We therefore present two different algorithms for placing multiple labels. The first method is a greedy approach and the second method shifts labels on the screen in order to place many labels with few occlusions.

5.3.1. Greedy Approach

A common approach for solving complex optimization problems as a combinatorial problem is using a greedy algorithm [FLH*06, SD08, ČB10]. For our greedy optimization the function $E(r)$ is extended to $E(r,n)$ which takes into account a specific label n with $1 \leq n \leq N$. The labels are then processed sequentially (e.g. from $n = 1$ to $n = N$) whereas a label, which has already been placed cannot be moved anymore and is internally 'locked'. For the currently processed label n, the candidate position for ray r is selected where $E(r,n)$ is maximal and no conflicts occur with the labels which have already been placed. This approach can lead to problems for higher n due to the (M)-requirements. Therefore latter processed labels may not be placed at all or exhibit a small quality value $E(r,n)$ which causes bad label arrangements with typically long connection lines. As indicated above, a specific order for processing labels is needed, which can improve the result significantly. The following sorting criteria were tested:

- *Order by quality*: In each step the label n with the lowest sum $\sum_{r=0}^{R} E(r,n) | E(r,n) \geq 0$ is processed next whereas previously placed labels are treated as locked labels and the sum is updated continuously.
- *Order by angle*: The labels are sorted by the angle spanned by each label center, the viewport center and the midpoint of the right viewport border.
- *From out to in*: The labels which lie on the convex hull of all visible labels are processed first (in CW or CCW order); the labels on the convex hull of the remaining labels are processed next etc.
- *From in to out*: Opposite order of previous ordering.

5.3.2. Label Shifting

The new shifting method is inspired by the human approach of shifting labels around, until a valid arrangement is found. The main idea is that labels are successively added to the image whereas the newly added label receives the best position and all labels occluding the new label have to be shifted aside. This approach especially supports the use case where annotations are added during reading and with that fulfills the requirement B1. The algorithm for adding a new label to an existing scene with a number of occlusion-free placed labels is outlined below (see Fig. 4 in addition).

1. Determine optimal candidate position for the new label L_{new} independent of other unlocked labels.
2. Place label L_{new} at the chosen position.
3. Define L_{new} as current label L_{curr}.
4. Test whether L_{new} is occluded by neighboring labels. Consider the nearest label on the adaptive border in clockwise (L_{next}) and counterclockwise (L_{prev}) direction.
5. If no occlusions occur, then a valid position is found!
6. If a conflict is detected on only one side, then:
   a. Determine shifting direction: shift CW if conflict with L_{next} or shift CCW if conflict with L_{prev}.
   b. *Repeat*: Shift label which occludes L_{curr} in the determined shift direction until this conflict is solved and the shifted label reaches a valid position regarding the (S)-requirements. Define shifted label as L_{curr}.
   c. *Until*:
      - no conflicts remain and thus a occlusion free arrangement is found.
      - a valid position cannot be found for L_{curr}.
      - L_{curr} is again L_{new} then no occlusion-free arrangement can be found.
7. If conflicts are detected on both sides, then:
   a. Apply steps 6b /6c with both shifting directions successively (L_{new} is not moved).

This algorithm entirely fulfills requirement M1 whereas label boxes may not overlap each other. But due to the shifting of labels over longer distances, intersections between connection lines may occur (see Fig. 5 top). In the following step, pairs of labels which produce such a *line conflict* have to be interchanged with regard to their label positions (more precisely the specific ray index). Unfortunately there are several problems which can arise from this simple procedure due to the different label sizes. After the interchange procedure labels may be assigned to invalid positions regarding the (S)-requirements or other pairs of labels may overlap each other (see Fig. 5 bottom). An idea for solving these issues is a concluding relaxation step where labels with conflicts are slightly shifted aside.

6. Evaluation of Algorithms and Weights

We evaluated the different settings of our label placement system with the goal to identify applicable algorithms and advisable weights of optional requirements for a robust and pleasant placing of labels. In the following we present the setup of our evaluation and the results including advantages and disadvantages of the various algorithms.

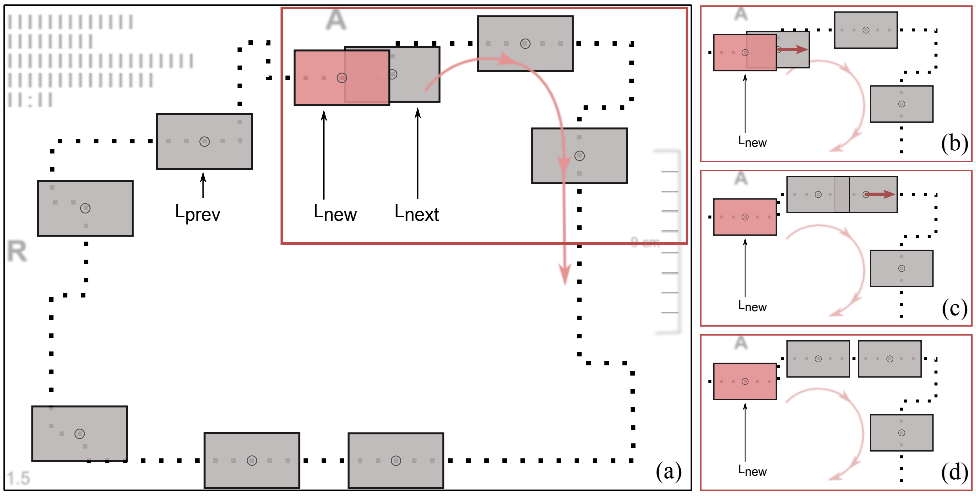


Figure 4: *Shifting approach: (a) The red marked label L_{new} is placed at the best candidate position. (b) The label L_{next} overlaps L_{new} and has to be shifted clockwise along the adaptive border to the next candidate positions. (c) After solving this conflict further label conflicts may occur. (d) New conflicts can be solved in the same manner until a valid arrangement is constructed.*

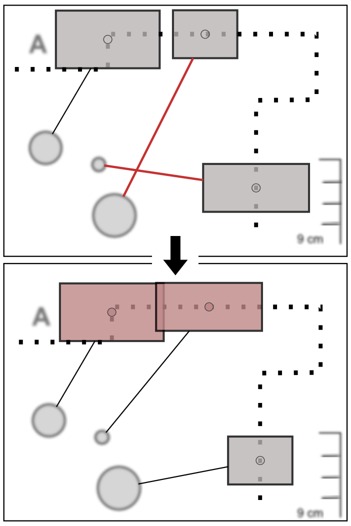


Figure 5: *Solving line intersections after adding all labels by shifting. Top: Possible label arrangement after adding labels. Bottom: Requirement M2 can be fulfilled by interchanging label positions of conflicting labels. This can lead to new conflicts due to overlapping labels.*

6.1. Setup

In addition to the shifting and the variations of the greedy algorithm, we also implemented two other approaches for the purpose of comparison. This includes variations of the exhaustive search method by using breadth- and depth-first search strategies on a graph containing the R^N possible label arrangements. The second approach uses the open-source library NLopt [Joh10] to solve a non-linear optimization problem which reflects the problem of minimizing the total line length under the constraint of a valid label arrangement.

Another aim is to find a set of pre-defined weights for the optional requirements which create a pleasant label arrangement. Therefore the weights γ_j for S5, S4, S1 and S3(ii) have to be evaluated whereas each weight can take a value between 0 (no effect) and 10 (high importance). For simplification we analyzed twelve combinations of weights: (0,0,0,0), (1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1), (1,1,1,1), (1,1,4,3), (2,1,1,2), (2,1,2,4), (3,1,3,2), (3,4,1,1), (4,1,2,3).

To assess the quality of the different label placement algorithms and weightings, we set up a framework where the label placement result of all four algorithms can be directly compared. To find out the best parameterization of each algorithm, 131 different views of abdominal datasets with 2-7 findings per view and different zoom levels and viewport sizes were recorded. For an objective evaluation we compared the following measures:

- computation time (at most two attempts taking each max. five seconds with shortened labels in the second attempt)
- placing ability (could all labels with placed with respect to constraints and full label content?)
- total length of connection lines (sum of line lengths)
- relative overlap between labels and image (how much label area of all labels overlaps acquired image?)

6.2. Results

After evaluation we ascertained that convenient algorithms regarding the trade-off between computation time and the

Algorithm	$\varnothing T1$ in ms	$\varnothing T2$ in ms	Placing Ability
Shifting	15,04	11,94	74%
Greedy (Quality)	18,42	12,7	60%
Optimization	749,10	741,55	(74%)
Breadth-first search	2956,63	377,75	42%
Depth-first search	634,85	170,08	76%

Table 1: *Extract of evaluation results: T2 is the average time needed in those views where all labels could be placed successfully with full content, T1 considers all tested views. The placing ability describes the percentage of views where all labels could be placed with full content.*

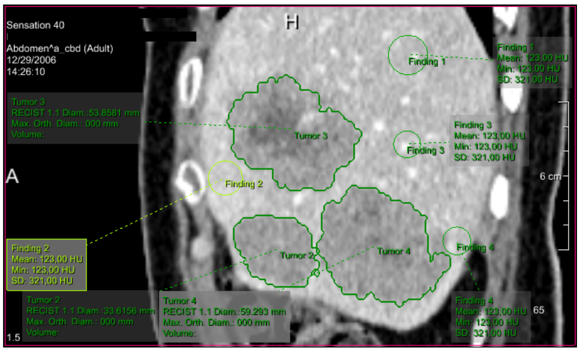


Figure 6: *Placement of seven labels in a viewport showing a liver via the shifting approach.*

ability to place many full-sized labels are shifting, greedy(by quality) and depth-search. The other tested algorithms are too slow for user interaction or place the labels disadvantageous such that not all labels can be placed considering the given requirements.

We noticed that the correlation between weighting and expected overall label arrangement is not very high for many findings due to the shifting of labels or the static character of the greedy method. However, good weight vectors (γ_{S5},γ_{S4},γ_{S1},$\gamma_{S3(ii)}$) for fewer findings regarding line length, relative image overlap and label placement ability are e.g. (4,1,2,3) and (3,1,3,2).

A main issue which affects the requirement G1 is that the result is heavily dependent on viewport size and zoom level, which also complicates the evaluation (more than ten labels can easily be placed in e.g. full screen viewports). For difficult views and many findings the exhaustive search has unacceptable time rates and optimization is also not applicable due to unpredictable computation time. Table 1 shows the average computation time and placing ability for the different algorithms (The optimization uses results of the shifting algorithm for initialization and thus the placing ability is mainly the same). The main advantages of the greedy approach are the fast computation and the good control of the quality of the position of single labels due to the processing order. Moreover, the requirement weighting has a higher effect on the overall label arrangement as the shifting algorithm which can only control the initial position of newly added labels. On the other hand, in many cases the greedy algorithm is not able to place all labels with full content due to unnecessary long connection lines or spacing between labels and as mentioned before, the result is highly dependant on the processing order. The shifting algorithm is overall the best for placing many labels with its full content at short computation times. It also fits best to the clinical task of placing labels during reading because of the very intuitive placement of the label of newly added annotations. Yet slightly overlapping labels or occlusion of findings by labels have to be accepted due to the interchange of label positions.

7. Medical Labeling System

In the following we want to demonstrate the application of the presented algorithms under consideration of the (B)-requirements. Therefore we merged the algorithms in a *Medical Labeling System* and observed its performance in a test environment where mainly abdominal CT data can be annotated by the user during a manually simulated reading process. However the algorithms are designed to be independent of the image modality or body region. The specific behavior of the labeling system can also be adjusted to test different configurations. The result of placing multiple labels in a viewport with the shifting algorithm is shown in Figure 6.

Adding annotations: Manual added annotations provoke the placement of a new label. The shifting approach can therefore be used whereas the newly added label gets the best position, in order to meet the expectations of the user, and overlapping labels are shifted aside. Thereby animated transitions between the label arrangement before and after the insertion ease the cognitive process of tracking labels.

Locking and unlocking labels: Labels are locked if the user moves the label. It is then excluded from the automatic placing process and the user is in charge of placing this label. After unlocking a label (right click → unlock), it is either integrated in the current label arrangement via greedy placing, via shifting at the next repositioning demand or an instant repositioning (via shifting) of all visible labels is induced.

Navigation (pan, zoom, rotate, slicing via right mouse button): While exploration, the displayed views may change significantly in few consecutive frames. To reduce flickering of labels, the complete label arrangement is only computed at mouse release. While the mouse is pressed down, the labels are either hidden completely or only labels which were visible at mouse press are displayed at their last computed position. This allows an uncomplicated rough search in the dataset without distraction.

Navigation (slicing via mouse wheel): For an accurate search, scrolling through the slices allows the user to choose

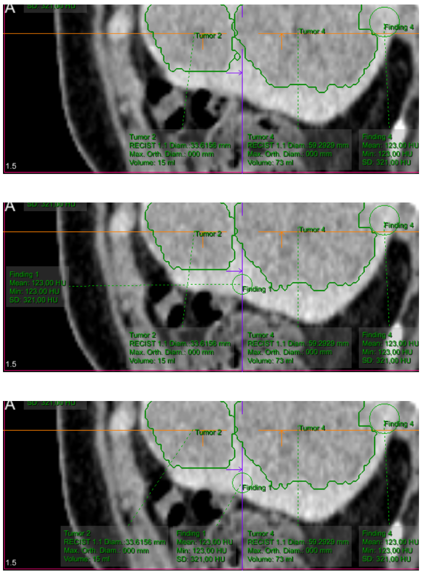


Figure 7: *Behavior of labels appearing during slicing via mouse wheel. Top: Arrangement before slicing. Center: Greedy placement of newly appeared label. Bottom: Repositioning of all labels with the shifting algorithm.*

a specific slice more precisely. Therefore the labels are always visible. If new findings appear, their labels are placed greedily without changing the position of recent visible labels. This is described as coherency and is important to guarantee readability by avoiding unnecessary movement of labels and to facilitate the tracking of single labels over multiple subsequent slices (see Fig. 7).

8. Conclusion and Future Work

The task of placing multiple labels in 2D slices of clinical applications is very complex due to the variable sized labels, the limited viewport size and the amount of constraints which are needed to guarantee a readable and pleasant label arrangement. Moreover the behavior of labels during user interaction has to be adapted precisely to the clinical workflow to support health professionals with additional information on the focused pathologies. We therefore presented two new approaches which improve the constrained labeling of 2D slice data. The reduction of the search space to a limited number of positions on the adaptive border is the basis for a fast evaluation of candidate label positions. Thereby many constraints can be integrated to optimally place single labels. The main challenge arises from placing multiple labels regarding readability and unambiguity. The new developed shifting algorithm assures a good utilization of free space and is able to place many labels also in small viewports. For the clinical application we combined the shifting and a greedy algorithm in a complex labeling system to achieve a seamless integration of labels in the clinical workflow such as reading and follow-up investigations.

We aim at improving the shifting algorithm such that overlaps resulting from label swapping are minimized. Further analysis of post-processing steps to enhance the quality of the overall label arrangement, e.g. a relaxation step which spreads the labels by diffusion, is thereby necessary. Another aspect is the performance of an extensive survey with medical experts and further investigations on requirement weights and the integration of algorithms in the clinical workflow.

References

[AHS05] ALI K., HARTMANN K., STROTHOTTE T.: Label layout for interactive 3d illustrations. *Journal of the WSCG 13*, 1 (2005), 1–8. 1, 2

[BF00] BELL B., FEINER S.: Dynamic space management for user interfaces. In *Proc. of the ACM symposium on User Interface Software and Technology* (2000), pp. 239–248. 2

[BFH01] BELL B., FEINER S., HÖLLERER T.: View management for virtual and augmented reality. In *Proc. of the ACM symposium on User Interface Software and Technology* (2001), pp. 101–110. 2

[ČB10] ČMOLÍK L., BITTNER J.: Layout-aware optimization for interactive labeling of 3D models. *Computers & Graphics 34*, 4 (2010), 378–387. 2, 5

[CMS95] CHRISTENSEN J., MARKS J., SHIEBER S.: An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics 14*, 3 (1995), 203–232. 2

[FLH*06] FUCHS G., LUBOSCHIK M., HARTMANN K., ALI K., STROTHOTTE T., SCHUMANN H.: Adaptive labeling for interactive mobile information systems. In *Proc. of Information Visualization* (2006), pp. 453–459. 2, 5

[HAS04] HARTMANN K., ALI K., STROTHOTTE T.: Floating labels: Applying dynamic potential fields for label layout. In *Proc. of Smart Graphics* (2004), pp. 101–113. 2

[Joh10] JOHNSON S. G.: The nlopt nonlinear-optimization package, June 2010. URL: http://ab-initio.mit.edu/nlopt. 6

[MD08] MAASS S., DÖLLNER J.: Seamless integration of labels into interactive virtual 3D environments using parameterized hulls. In *Proc. of Computational Aesthetics in Graphics, Visualization, and Imaging* (2008), pp. 33–40. 2

[MP09] MÜHLER K., PREIM B.: Automatic textual annotation for surgical planning. In *Proc. of Vision, Modeling, and Visualization* (2009), pp. 277–284. 1

[MS91] MARKS J., SHIEBER S.: *The Computational Complexity of Cartographic Label Placement*. Technical report 05-91, Harvard University, 1991. 2

[RPRH07] ROPINSKI T., PRASSNI J.-S., ROTERS J., HINRICHS K.: Internal labels as shape cues for medical illustration. In *Proc. of Vision, Modeling, and Visualization* (2007), pp. 203–212. 1, 2

[SD08] STEIN T., DÉCORET X.: Dynamic label placement for improved interactive exploration. In *Proc. of Non-photorealistic animation and rendering* (2008), pp. 15–21. 2, 5

[TAE*00] THERASSE P., ARBUCK S. G., EISENHAUER E. A., WANDERS J., KAPLAN R. S., RUBINSTEIN L., VERWEIJ J., VAN GLABBEKE M., VAN OOSTEROM A. T., CHRISTIAN M. C., GWYTHER S. G.: New guidelines to evaluate the response to treatment in solid tumors. *Journal of the National Cancer Institute 92*, 3 (2000), 205–216. 1, 3