Society for Science and Education
United Kingdom

JBEMi Journal of Biomedical Engineering and Medical Imaging

ISSN: 2055-1266
Volume 5 Issue 2

External Group Labeling of Objects in 2D Medical Images Using Spring-Mass Model

[1]Muhammad Masud Tarek, [2]Bilkis Jamal Ferdosi
[1]Associate Professor, Department of CSE,
State University of Bangladesh, Dhaka, Bangladesh
[2]Associate Professor, Department of CSE,
University of Asia Pacific, Dhaka, Bangladesh
tarek@sub.edu.bd; bjferdosi@uap-bd.edu;

ABSTRACT

The parallel processing of verbal elements like textual labels, annotations etc. with non-verbal i.e. visual elements, provides maximum integration of human mind. Textual labels and annotations aid in understanding medical data visualization. But it is a very challenging task to generate hand-made like illustration labels in an automated system. The paper presents a noble approach for dynamic external labeling of segmented 2D computer tomography (CT) slices. Various efficient algorithms are approached to solve labeling problems in modular form. The system emphasizes on the readability and clarity of labels while generating dynamic group labels in real time for 2D slices using external labeling layout.

Keywords: External labels, medical data visualization.

1 Introduction

Images are considered as a very powerful medium for communicating ideas. However to make an image more meaningful and informative, textual descriptions such as labels, legends, captions etc. have become essential parts of images, especially in educational and scientific documents. Labels are largely classified as internal and external labels. Internal labels are placed within the objects of a picture. On the other hand, external labels are placed into the empty spaces, outside of the objects. Both internal and external labels are used to describe or distinguish different graphical elements. In case of external labels, lines are drawn between labels and their corresponding referenced objects. As aesthetic is a very important factor, illustrators have to be very careful to maintain the visual balance while labeling a handmade illustration. Automated and interactive labeling systems are still a challenge. Poor placement of labels may have a negative impact on understanding a visual system. Whether it is an interactive or automated system, the main goal of labeling an illustration is to employ any of the layout conventions that are used by human illustrators, so that labels can be read comfortably and visual balance of the image is preserved.

For quantitative measurement, segmentation of computer tomography (CT) is needed by many image analysis procedures. Research are going on to produce CT segmented images [12, 13]. Different segmentation masks are used to distinguish background and different structures. In real life application,

DOI: 10.14738/jbemi.52.4445
Publication Date: 29th April 2018
URL: http://dx.doi.org/10.14738/jbemi. 52.4445

doctors sometimes need a verbal description of the segmented structures. Also, medical students need textual contents along with the CT images for better understanding. Manual labeling would be a naïve way for this type of application as hundreds of labels might need to be generated on the fly during analyzing CT images slice by slice. So here, a dynamic label placement system is developed which can generate external textual labels for segmented CT images. The system finds good placements of labels while viewing 2D slices with the consideration of different labeling aspects.

In section 2, some related works are reviewed. Section 3 states the system architecture while section 4 describes the developed dynamic labeling system. Section 5 provides some results with the discussion. Finally, the paper concludes with some hints of future extension.

2 Related Works

Dynamic textual labeling for CT images is still largely unexplored. However, there are some interactive systems for 3D models that address some of the labeling problems. For example, Zoom Illustrator [15] uses dedicated part on the screen for text so that they do not overlap any image object. Also, Fish View Technique [8] is used to show more textual information without overlapping other labels. Talking Shadows [4] and Illustrative Shadows [16] use object shadows for placing text information of the associated objects. But they hardly address any labeling layout problem.

A Virtual Reality application, View Management [3] dynamically places labels for objects. It considers most of the labeling problems such as internal or external labeling layout, empty space management, minimization of label overlapping etc. However, as it uses bounding box technique to determine the area of an object, thin long diagonal objects may wrongly occlude other small objects. Map labeling is a well-studied field and many works are done for dynamic internal map labeling layout. However, it is found that global optimization (i.e. no overlapping of labels) is computationally N-P hard [6]. So, change of one label position effects the entire map labeling layout.

Few good algorithms are developed for 3D dynamic labeling. They consider major labeling layout problems such as label layouts (flash or radial, internal or external), placement of external labels into empty space, minimization of line crossings, elimination of label overlapping etc. Floating Labels [10] uses some potential force fields to eliminate overlapping of labels. It also addresses frame coherency to minimize visual discontinuity. In [9], calmer parts of the animated 3D objects are determined where anchor points or internal labels can be put to achieve lesser flickers among frames. Ali et al. [1] employed different layout methods for external labeling of 3D illustrations. The system is greatly suitable for compact and convex 3D models with having enough empty spaces surrounding the object. These techniques can be adopted for 2D layered CT images with considerations. However, achieving all the aesthetic criteria of a labeling layout at the same time is very difficult as some criteria may conflict each other. Also most approaches trade-off between labeling quality and computational costs.

Oeltze-Jafra and Preim surveyed several techniques of labeling in medical visualization [20]. They proposed a categorization of different labeling techniques and recommended guidelines for choosing a proper technique for a specific type of application area. They indicated that it is a hard task to achieve all requirements of labeling. They also reported that among several techniques labeling slice views is a relatively less explored area.

In [21] Kang et al. proposed a force-directed labeling method for 3D street visualization that increases the visibility of street addresses in the map. However they addressed the problem of internal labeling whereas our method address the problem of internal labeling.

3 System Architecture

After reviewing related works and analyzing medical illustrations [17, 19] it is found that a good labeling system should consider the following aspects:

Readability: Labels and lines should have good background contrast. Font size, text shadow, background and foreground color can be adjusted by the users as needed.

Aestheticism: Placement of the labels should be such that there should not be any overlapped label. Also, line crossings should be eliminated or reduced. Anchor points should be at salient positions of the structures.

Efficiency: Good, efficient algorithm should be used so that the system can be implemented for real-time application.

Grouping: Same structure may be seen in different places. Some criteria should be utilized to group them together and thus minimizes the number of labels.

Slice Coherency: Sizes, locations of structures may vary within different slices. So it is likely to vary the position of labels also, which may generate the blinking effect. So measures should be taken to minimize this blinking effect.

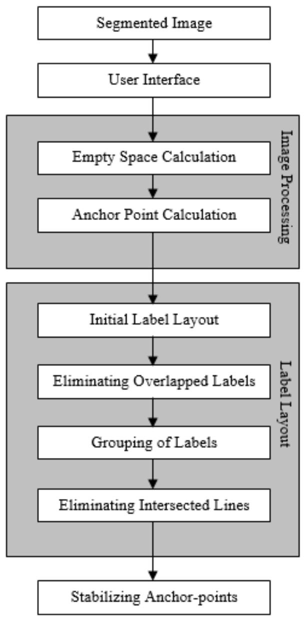


Figure 1: System Architecture

The system architecture (Fig. 1) is designed to achieve the above mentioned global requirements. The input segmented image itself contains the identification text (labels) and other color-coded information for each object, while users can define various label properties such as text size, color, text-background color, and transparency etc. in the *User Interface (UI)* module. In the *Image Processing* modules, available empty spaces (i.e. background) are calculated where labels can be put. Acceptable anchor points for each object are also calculated here. A textual label is connected to an anchor point by a line to indicate the corresponding object. In the label layout modules, initial placement of each label is determined. Here, there are several other modules for eliminating label overlapping, grouping and eliminating intersection of connected line segments and thus final placements of all the labels of a slice are determined. The last module stabilizes anchor points to reduce visual discontinuity between two adjacent slices.

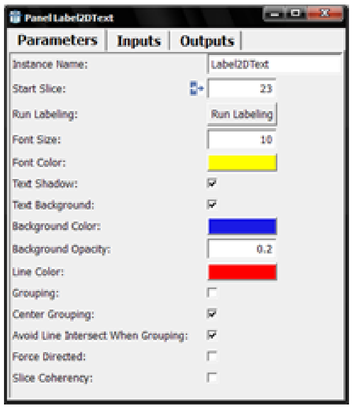
4 The Dynamic Labeling System

A modular approach is used to implement the dynamic labeling system. For the better contrast between image and textual labels, each label is enclosed within a rectangular box. The size of the box depends on the text length and font size of the label. Users are able to change rectangle background color on the fly (Fig. 2). For the aesthetic reason, these rectangle boxes are transparent, so that some parts of the graphical structures are visualized and thus minimizing visual clutter. Alpha Blending [14] is used for the transparency effect using the equation 1:

$$\begin{aligned}&NewPixelValue = BG*(1-\alpha)+FG*(\alpha)\\&where\ BG = backgound\ pixel\ value\\&FG = foregound\ pixel\ value\\&\alpha = 0 \text{ to } 1\end{aligned} \tag{1}$$

For each slice, the generalized algorithm is:

1. Calculate background spaces which are available for putting labels
2. Identify each object uniquely
3. Calculate anchor point for each object
4. Determine initial label placement
5. Eliminate label overlapping by either
    a. moving overlapped labels up or down repetitively to a minimum distance, or
    b. using potential repulsive forces among labels and attractive forces between an anchor point and corresponding textual label of an object
6. Group same objects under a single textual label if those objects are within a threshold distance
7. Draw connection line from anchor points to their associated label
8. Eliminate connected line intersections
9. Stabilize anchor points of next slice if they are within a threshold value of current slice.



Instance Name:	Label2DText
Start Slice:	23
Run Labeling:	Run Labeling
Font Size:	10
Font Color:	🟨
Text Shadow:	✓
Text Background:	✓
Background Color:	🟦
Background Opacity:	0.2
Line Color:	🟥
Grouping:	☐
Center Grouping:	✓
Avoid Line Intersect When Grouping:	✓
Force Directed:	☐
Slice Coherency:	☐

Figure 2: User Interface

4.1 Uniquely Identifying an Object

In a segmented CT slice, objects are color-coded uniquely but the same object (i.e. same color code) may appear in different places within a slice. For a successful labeling, each part has to be uniquely identified. Connected component labeling algorithm [5] may be used for that purpose. But if objects have many edges, a sheer amount of data in the equivalent table have to be managed which is extremely difficult. So here the system scans 8-neighbors of a pixel recursively to determine the area of an object. The algorithm is fast enough for a real-time system as a flag variable is used to avoid additional scanning of a pixel.

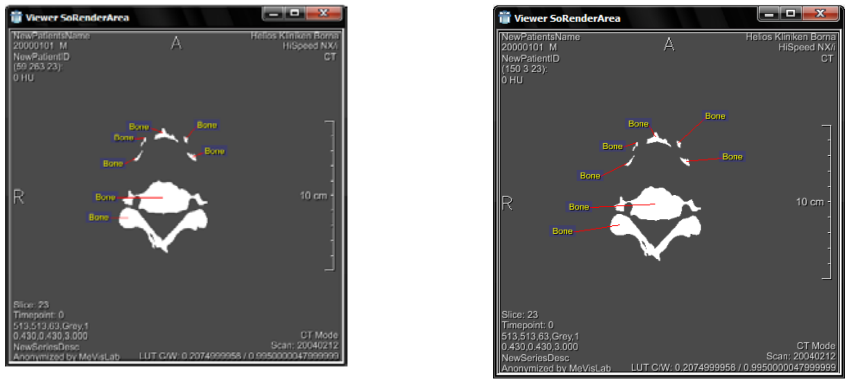
4.2 Anchor Point Calculation

The point of the object up to which the line from a textual label is drawn is known as an anchor point. As placement of a label depends upon anchor point position, a good calculation is much needed. For a fast output, any point of an object could be used as an anchor point. But it would not be an aesthetic choice. In [15], some criteria are discussed that should be taken into account during anchor point computation. For example, the point should be valid i.e. part of the corresponding structure. To avoid overlapping and visual clutter, anchor points should not cluster together and of course, calculations should be fast enough for an interactive system. Most importantly, it should be a salient point, i.e. at the visually most dominant part of the object. Skeletonization [11] is a good approach to find a salient point but computationally very expensive $O(n^2)$. To meet most of the criteria, the system uses the most inner point of an object as the anchor point using distance transform mapping [18] with computational cost $O(n)$. To avoid real number calculation, Euclidean or city-block distance matrices are avoided. Instead, Pseudo-Euclidean method [2] is implemented which uses integer approximation. It would not give actual distance (which is not required for this system) but the most inner point of an object can be determined efficiently.

4.3 Label Placement

For simplicity, the system uses flash left-right labeling layout. So, a label will be placed either left or right side of an object. Initially, after sorting all the anchor points of a slice, the system would try to place labels from top to bottom one by one. If the anchor point is at the left side of the slice, the system would try to find enough empty space at the left side of the object for the label. If failed, it would place on the other side. Initially, a label is placed at the same y-coordinate of the associated anchor point. If any two labels are overlapped, they would be moved up or down at a minimum distance. The value of the minimum distance can be chosen by the user. Other already placed labels would be rechecked repeatedly to ensure no overlapping is happened (Fig. 3a).

The system can also use force-directed method for the placement of labels. Fruchterman and Reingold introduced a spring-embedding approach for graph drawing which considers two forces [7]. The attractive force attracts the connected vertices each other while the repulsive force causes all other vertices to repel each other. The same approach can be used for the external labels. Here a label is attracted by associated anchor point, whereas, labels have repulsive forces to each other. To avoid the collapse of a label to its object, all the pixels of that object except anchor point have repulsive forces on the label. Also, labels get repulsion from image boundary walls so that they will always be within the image area.



(a) (b)

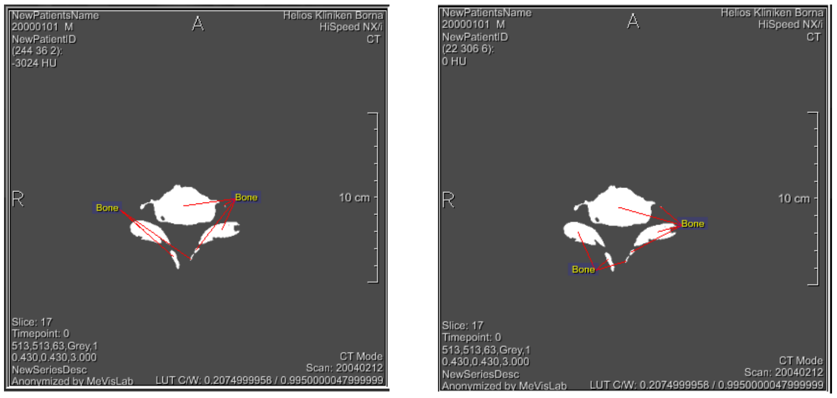
Figure 3: (a) Without force-directed method labels are put closer, (b) Force directed approach puts labels more uniformly

To employ the graph drawing method, labels are considered as zero length points. The attractive force between an anchor point and associated label can be given by,

$F_a(x) = x^2/k$ where x is the distance between the anchor point and the label and k is the tolerable distance between them. Typically the value of k is 30 to 50. Similarly, repulsion forces among labels can be calculated by,

$F_r(x) = -k^2/x$ where x is the distance between two labels and k is the ideal distance between them.

All forces are summed up to calculate the displacement for a label (Fig. 3b). However, a temperature t is used to control the displacement. At first, the temperature is given high (1.0) and a cooling function is used to cool it in each iteration. The typical cooling rate is 0.95 to 0.90. During iteration minimum of temperature and normalized force value is used for a small displacement of the label. Typically, after 20 to 30 iteration a satisfactory label placement can be achieved (Algorithm 1).



(a) (b)

Figure 4: Grouping (a) without and (b) with considering the centroid of anchor points.

4.4 Grouping

In a CT slice, same object (or structure) may appear more than once. Sometimes it is desirable to group those objects under one label. This would make the output image more readable. However, it may not always produce a good result. Many objects under one label may cause visual clutter. So, for grouping of labels, we consider some criteria such as the number of maximal objects that can be grouped together, maximum distances among group objects etc. The real challenge is the good placement of label for a group of objects. One simple approach is to take the initial label position of the most upper object of the group. But it would not give a salient position. So the system uses more complex approach. Here, the centroid of the candidate anchor points is calculated. The initial label of the nearest anchor point of the centroid is served as the label of all group members. All other labels are deleted and connection lines are redrawn from the selected label to anchor points (Fig. 4).

Algorithm 1: Force-directed method

```
L = list of labels
A = anchor points
W = boundary points of the slice
O = list of objects' border points

for iteration:=1 to max_iteration // typically, max_iteration=20 to 30
 for each L[i]
     TDisp[i]:=0
     // a label attracts its referenced anchor point
     Disp:=L[i]-A[i]
     TDisp[i]:= TDisp[i]- sign(Disp)* AForce(abs(Dis))

     for each L[j]
         if (i!=j)
             // a label repulses other labels
             Disp:=L[i]-L[j]
             TDisp[i]:= TDisp[i]+ sign(Disp)* RForce(abs(Disp))
```

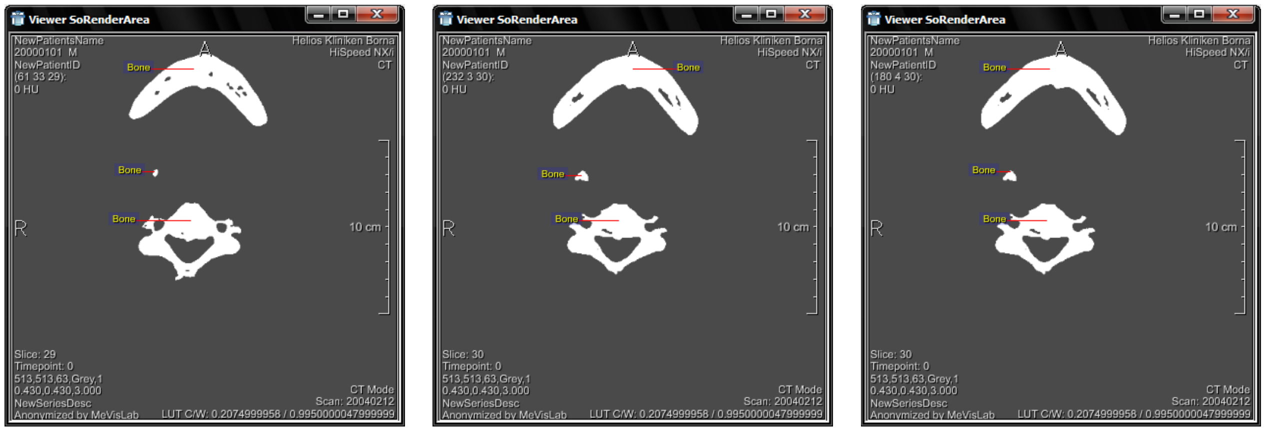
```
                // a label repulses non-referenced anchor points
                Disp:=L[i]-A[j]
                TDisp[i]:= TDisp[i]+ sign(Disp)* RForce(abs(Disp))
            end if
        end for
        // a label repulses slice-border points
        for each W[j]
            Disp:=L[i]-W[j]
            TDisp[i]:= TDisp[i]+ sign(Disp)* RForce(abs(Disp))
        end for
        // a label repulses its object's border points
        for each O[i]
            Disp:=L[i]-O[i]
            TDisp[i]:= TDisp[i]+ sign(Disp)* RForce(abs(Disp))
        end for
        // calculate new label position with initially, Temperature=1
        EffectiveDisp:= sign(TDisp[i]) * min(norm(TDisp), Temp)
        L[i].pos:=L[i].pos + EffectiveDisp
        Temp=cool(Temp)
    end for
end for

function AForce(x){return  $k^2$/x;} // k=ideal distance
function RForce(x){return $x^2$/k;}// k=tolerable distance
function cool(temperature) {return temperature * coolingRate;}
     //initially temperature=1.0, typical coolingRate=0.95
```

4.5 Eliminating Line Intersections

Using elementary geometry knowledge, each pair of line segments are checked for any intersections. For non-group labels, if any intersection is found, label positions are swapped, thus eliminating line intersections. After swapping, corresponding segment is again re-checked with other lines. As a number of labels within a slice are not much (usually less than 30), so the above-mentioned method can be used in real time system without much difficulty. For grouped labels, if any intersection occurs, it would be more complex to solve. In that case, the object which is responsible for line intersection is left out from the group.

4.6 Slice Coherency

The system generates labels dynamically according to objects' location, computed anchor points and available empty spaces. When users go through slices interactively, it is more desirable to have a minimal change of label locations. The abrupt change of labels may generate flicker and users' attention may be destructed. If users go through slices very quickly, as new anchor points are likely to generate in every slice due to change of object's size and location. As a result, blinking effect may produce. To reduce this effect, anchor points should not be changed as long as possible. However, this may cause non-salient anchor points for objects. The system remembers the anchor points of the previous slice. After computing new anchor points for current slice, it compares them with previous positions. If positions are not changes for a certain thresh-hold value, previous positions are used. Label positions are unlikely to change much for unchanged anchor points. As a result, labels do not jump much (Fig. 5).



(a) (b) (c)

Figure. 5: With no slice coherency, slice 29 (a) and slice 30 (b) have different label positions, while, with anchor point stabilization approach, slice 29 (a) and slice 30 (c) have almost similar label positions thus minimizes blinking effects.

5 Results and Discussion

The system is developed using MeVisLab, a medical data visualization tool based on QT framework and Open Inventor toolkits. The primary goal of the system is to generate external labels for CT slices which would help doctors and medical students to recognize different objects within the slices easily. So, for such system, perfect evolution should be empirical, i.e. users' feedback. Unfortunately, as the system is still in developing phase, no real-world user feedback is available. From the numerical point of view, four segmented CT images were used to generate labels with total 154 slices (Table 1). Among 1074 objects, all were labeled successfully. Because of attractive forces among objects and anchor points, it is seen that labels overlapped with objects (2.14%) are slightly more than while no force-directed method is used (1.49%). Though, both figures are very low, if compare to a number of total objects. There is only 0.1% line intersection which is a great result for readability and clarity. As there are usually less than 30 objects per slice, current standard computers take less than 2 seconds to generate labels for a slice. Even force-directed method takes insignificant time. So, the system is very suitable for real-time application.

Table 1: Summary of experimental results

No. of CT images = 4 Total slices =154 Total objects =1074 Failure to label = 0
With Non-grouping labeling No. of label overlap (no force)=12 No. of line Intersect (no force)=02 No. of label overlap (with force)=16 No. of line Intersect (with force)=03
With grouping labeling No. of line intersect=02 No. of label overlap each other=0 No. of label overlap with objects (no force) = 16 (1.49%) No. of label overlap with objects (with force) = 23 (2.14%)

6 Future Work and Conclusion

Generating textual labels dynamically for CT slices are very helpful for both doctors and medical students. As usual, there are always some scopes for future extensions of the work. The system currently only uses left-right flash layout. It can be extended for other labeling layouts like top-bottom, ring, radial etc. The system uses 2D texts for labeling. Using 3D texts in future more features like text zooming, rotating etc. can be added. Another good extension would be contextual grouping where objects with similar functionalities can be labeled together. Interactivity can be added so that users can choose which functional objects will be labeled.

ACKNOWLEDGMENT

The author thanks Prof. Bernhard Preim, Faculty of Informatics, OvG University of Magdeburg, Germany for lab facilities and input source images.

REFERENCES

[1] K. Ali, K. Hartmann, & T. Strothotte*, "Label layout for interactive 3D illustrations"* in Journal of the WSCG, 13:1–8, 2005.

[2] Carlo Arcelli and Gabriella Sanniti di Baja*, "Finding local maxima in a pseudo-euclidean distance transform"* in Computer Vision, Graphics and Image Processing, 43(3):361–367, 1988.

[3] B. Bell and S. Feiner, *"Dynamic space management for user interfaces"* in Proceedings of Symposium on User Interface Software and Technology, p 238–248, 2000.

[4] W. Chigona, H. Sonnet, F. Ritter, and T. Strothotte, *"Shadows with a message"* in Smart Graphics: Third International Symposium on Smart Graphics 2003, pp. 91–101. Springer Verlag, Berlin, 2003.

[5] M. B. Dillencourt, H. Samet, and M. Tamminen, *“A general approach to connected-component labeling for arbitrary image representations”* in J. ACM 39(2), pp. 253–280, 1992.

[6] M. Formann and F. Wagner. *“A packing problem with applications to lettering of maps”* in ACM Symp. on Comp. Geometry, volume 7, pages 281–288, 1991.

[7] T.M.J. Fruchterman and E.M. Reingold*, “Graph drawing by force-directed placement”* in Software-Practice and Experience, 21(11):1129– 1164, 1991.

[8] Furnas, G.W. *"Generalized fisheye views”*, Proc. ACM SIGCHI'86 Conference on Human Factors in Computing Systems, Boston, April, pp. 16-23, 1986

[9] T. Götzelmann, K. Hartmann, and T. Strothotte *"Annotation of animated 3D objects"* in Online Proc. 18th Simulation and Visualization Conference, Magdeburg, Germany, March 2007

[10] K. Hartmann, K. Ali, and T. Strothotte, *“Floating Labels: applying dynamic potential fields for label layout”* in Smart Graphics: 4th International Symposium (SG 2004), pp. 101–113, 2004.

[11] Ramesh Jain, Rangachar Kasturi, and Brian G. Schunck in Machine Vision, McGraw-Hill, Inc., 1995.

[12] S. Loncaric and D. Kovacevic, *“A method for segmentation of CT head images”* in Proceedings of the 9th International Conference on Image Analysis and Processing, pp.388-395, Fl., Italy, 1997

[13] Z. Majcenic and S. Loncaric, *“CT image labeling using simulated annealing algorithm”* in Proceedings of the IX European Signal Processing Conference, Vol. 4, pp. 2513-2516, Island of Rhodos, Greece, 1998.

[14] Thomas Porter and Tom Duff, *“Compositing digital images”* in Computer Graphics, 18(3), July 1984, 253-259

[15] B. Preim, A. Raab, and Thomas Strothotte, *“Coherent zooming of illustrations with 3D-graphics and text”* in Proceedings of Graphics Interface, pages 105–113, 1997.

[16] F. Ritter, H. Sonnet, K. Hartmann, and Th. Strothotte, *“Illustrative Shadows: integrating 3D and 2D information displays”* in International Conference on Intelligent User Interfaces, pages 166–173, 2003.

[17] W. Rogers, Textbook of Anatomy, Churchill Livingstone, Edinburgh, 1992.

[18] Rosenfeld and J. Pfaltz*, “Distance functions in digital pictures”* in Pattern Recognition, vol 1, p33–61,1968.

[19] J. Sobotta, R. Putz, and R. Pabst, editors. *Sobotta: Atlas of Human Anatomy.* Volume 2: Thorax, Abdomen, Pelvis, Lower Limb. Williams & Wilkins, Baltimore, 12. English edition, 1997.

[20] S. Oeltze-Jafra and B. Preim. (2014). *Survey of labeling techniques in medical visualizations.* In Proceedings of the 4th Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM '14). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 199-208. DOI=http://dx.doi.org/10.2312/vcbm.20141192.

[21] Kang, Y.S., Park, S.H. & Joo, Y.J. Spat. Inf. Res. (2016) 24: 3. https://doi.org/10.1007/s41324-016-0003-4.