Floating Labels: Applying Dynamic Potential Fields for Label Layout

Knut Hartmann, Kamran Ali, and Thomas Strothotte

Department of Simulation and Graphics
Otto-von-Guericke University of Magdeburg
Universitätsplatz 2, D-39106 Magdeburg, Germany
{hartmann, kamran, tstr}@isg.cs.uni-magdeburg.de

Abstract. This paper introduces a new method to determine appealing placements of textual annotations for complex-shaped geometric models. It employs dynamic potential fields, which consider attractive and repulsive forces between pictorial elements and their textual labels. Several label candidates are computed and evaluated according to weighted penalty functions. The individual weights can be adjusted according to global design decisions or user preferences. Moreover, the user can re-arrange individual labels whereas the system re-adjusts the remaining labels. The method is demonstrated by the FLOATING LABEL system.

1 Introduction

A wide range of different application domains benefit from providing natural language explanations for pictorial elements. Frequently, they contain the denotation or unknown technical terms and are thus crucial to establish co-referential relations between pictorial and textual expressions. The relation between pictorial elements and their associated textual labels is indicated by meta-graphical objects such as connecting lines and anchor points placed on top of pictorial elements.

This concept is very attractive for interactive multi-modal information systems (e.g., ZOOM ILLUSTRATOR [15], ILLUSTRATIVE SHADOWS [16]). In these systems two- or three-dimensional geometric objects are associated with a set of labels. The content to be presented within labels depends on the interaction context and therefore its size changes dynamically. An appealing label layout has to balance various optimal criteria, and should be adjustable to superior design decisions. This paper develops a new technique to layout textual labels for complex-shaped geometric models, which balances various evaluation criteria and is easily adjustable.

We collected a corpus of illustrations with many textual labels, extracted from anatomic textbooks, anatomic atlases, pictorial dictionaries, and technical documentations (e.g., [7,17,5]). In a manual analysis we made some simple observations:

1. Labels should be placed near to their reference objects in order to prevent co-referential mismatches. They could be either placed within or outside their reference objects (internal vs. external placement).
2. The length of the line connecting label and pictorial element should be minimized.
3. The connecting line should be orthogonal to a main axis.

4. Crossings between connecting lines should be prevented.
5. The placement of the anchor point should ease the identification of the pictorial element.
6. In dynamic environments the label placement has to be coherent, i.e., the label position in time $i+1$ should be near to its position in time i.

The central idea to formalize these requirements is to establish positive and negative fields, which induce attractive and repulsive forces between pictorial elements and associated labels. This strategy is known as the *artificial potential field method*. Section 2 introduces this method. Section 3 formalizes the above observations in terms of attractive and repulsive forces, which are also exploited to evaluate various label placement candidates. Section 4 presents our initial results, and Section 5 discusses extensions. Finally, by analyzing related work in Section 6, Section 7 summarizes the new contributions of this work.

2 Potential Fields

Artificial potential fields have been introduced in Robotics by KHATIB [9] and became popular in other fields of Artificial Intelligence. They are inspired by attractive and repulsive forces between electric particles. Therefore, the terms (electric) potential, and its first derivative, the (electric) field, are used.

The classical textbook on robot motion planning by LATOMBE [10] contains a popular formalization, where attractors are modeled as global minima and retractors as maxima within the potential field. The algorithm calculates attractive and repulsive forces for each possible position p for a robot on a plane in a static environment. The attractive force steers the robot in the direction of the target:

$$U_{attr}(p) = c_1 \rho^2_{goal}(p)$$

where $\rho_{goal}(p)$ denotes the minimal distance from p to the target and c_1 is a scaling factor. Moreover, for each obstacle B_i a repulsive force is established:

$$U_{B_i}(p) = \begin{cases} c_2\left(\frac{1}{\rho_i(p)} - \frac{1}{\rho_0}\right)^2 & , \rho_i(p) \leq \rho_0 \\ 0 & , \rho_i(p) > \rho_0 \end{cases}$$

where $\rho_i(p)$ denotes the minimal distance of p to the obstacle B_i, ρ_0 the spatial influence of the obstacle, and c_2 is another scaling factor. The artificial potential field is defined as the sum of the attractive and all repulsive forces:

$$U(p) = U_{attr} + \sum_i U_{B_i}$$

The algorithm determines the forces at the initial position of the robot and iteratively redirects the robot according to the field F until it reaches a minimum within the potential field. Therefore, gradients $\overrightarrow{\nabla}$ of the potential field U on the position p are computed:

$$\overrightarrow{F} = -\overrightarrow{\nabla} U(p)$$

Throughout this paper we will refer to the objects, which float according to these forces as *particles*. Real applications often employ grid-based space quantization. The challenge in all these applications is to prevent those particles getting stuck within local minima without reaching the global minima.

3 The Floating Labels System

Some of the observations stated in Section 1 can be reformulated in terms of attractive and repulsive forces. Figure 1 depicts several forces, which form the basis of our FLOATING LABELS algorithm:

A: An attractive force between a pictorial element and its associated label,
B: a repulsive force at the object boundary (i.e., the label should be placed entirely within or outside its reference object),
C: a repulsive force between the label and all other pictorial elements,
D: repulsive forces between labels and the image boundary, and
E: repulsive forces between labels.

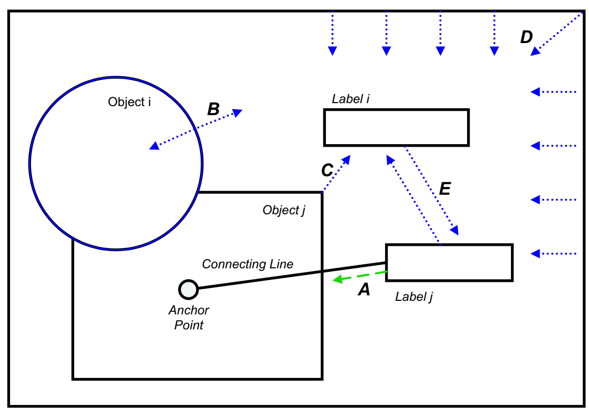

Fig. 1. The force system used by the FLOATING LABELS algorithm. Thick lines denote pictorial elements, associated labels and meta-graphical symbols (anchor points and connecting line). Dashed lines indicate attractive forces, and dotted lines show repulsive forces.

The formalization of potential fields in the previous section is based on the assumption of a single attractive force and constant repulsive forces. In contrast, the FLOATING LABELS system induces a potential field for each individual geometric object and its associated label and assumes dynamic floating labels, inducing dynamic repulsive forces. Note that the Forces A–D remain constant irrespective of the label configuration, whereas Force E is sensitive to the label configuration.

The *static potential field*, i.e., the accumulated attractive and repulsive Forces A–D, is used to determine a set of label placement candidates. Therefore, the algorithm inserts some label particles in the potential field, which then move according to the field F. To expand this point-feature label abstraction, the algorithm determines an area which contains the particle's position and which minimizes the accumulated potential over the label's area. The FLOATING LABELS algorithm considers the label's center and corners. After estimating the best candidates, the overlappings between these labels are resolved in the second phase. In the following we will present the architecture of the FLOATING LABELS system and the formalization of the various attractive and repulsive forces.

3.1 System Architecture

Figure 2 illustrates the architecture of the FLOATING LABELS system. In order to support the integration of various applications, a domain expert establishes the connection between geometric objects and terms in natural language. The domain expert contains or collects all information which is required to establish and exploit multi-modal co-referential relations. The FLOATING LABELS provides advices about an appealing label layout.

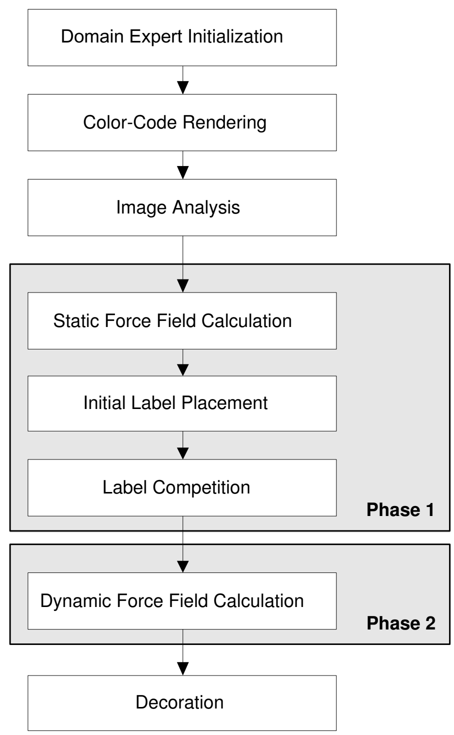

Fig. 2. Architecture of FLOATING LABEL system.

The FLOATING LABELS algorithm is based on color-coded projections of the scene displayed in a 3D application. Therefore, a color-code renderer informs the domain expert about the color encoding scheme, so that the color values could be used to determine visible objects and to retrieve the label's content.

The image analysis extracts the parameters required to compute the static potential field. The initial label placement determines the start positions of label particles, which then float to reach minima within the potential field. These point-feature label abstractions are then expanded to area features and evaluated according to a number of penalty functions. The best label placement candidates are then re-adjusted according to the dynamic potential field. Finally, the label coordinates are sent to the application and rendered (decoration).[1]

3.2 Image Analysis

Based on color-coded projections, the image analyzer determines visible objects. Due to occlusions, the projections of cohesive objects might be split into several segments. For each pictorial element the image analysis creates a list of all segments, their extent, size, and an internal point.

Determination of Internal Points: The placement of the anchor point is crucial to establish the co-referential relation between a pictorial element and its label. Independent from the geometric properties of the reference object, an anchor point has to be placed within the pictorial element. For convex objects, the center of mass is a good candidate. However, the segments could have arbitrary shapes and may contain holes. Therefore, we use a *thinning* algorithm [13], which iteratively shrinks the segment and stops as the segment completely disappears. The last collapsed pixel is the most internal one and is selected as anchor point. This location is furthest away from the silhouette and lies inside the thickest region. This guarantees that the anchor point is placed at a visual dominant region. We will refer to this point within an object O as $interia(O)$.

An analysis of the label layout of hand-made illustrations in our corpus reveals that frequently only one segment is labeled, i.e., only one pair of meta-graphical objects (anchor point, connecting line) establishes the co-referential relation between the pictorial element and its associated label. Currently, the anchor point is placed on an internal point of the biggest visible segment. However, the results of our first experiments suggest to use the segment with the minimal distance to the textual label.

3.3 Static Force Field Calculation

For each object a separate potential field is constructed, considering the object's area as an attractive field (Force A) and the area occupied by all other objects as repulsive force (Force C). Moreover, long-distance attractive forces ensure that each label is directed towards the center of the largest segment of the reference object.

A Laplacian edge detector is applied on the color-coded image. A repulsive function U_{silh} aims at placing the labels either within or outside their reference objects (Force B). Another repulsive function U_{wall} considers the minimal distance to the image borders, in order to prevent labels from leaving the image (Force D). Both repulsive functions need not to be recomputed for specific label configurations. In the following we will present the attractive and repulsive functions employed in the FLOATING LABELS system.

[1] An extended version of this paper also includes the complete layout algorithm (`http://wwwisg.cs.uni-magdeburg.de/ hartmann/Papers/floating-labels.pdf`).

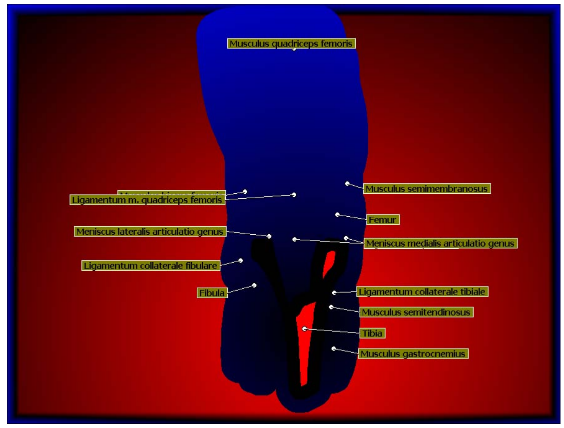

Fig. 3. A potential field for the tibia, a great bone located at the lower part of the knee joint. Red encodes attractive forces, and blue encodes repulsive forces. Black areas expose an equilibrium between attractive and repulsive forces.

The attractive force between a pictorial element and its associated labels is defined by:

$$U_{attr}(p) = \begin{cases} 0 & , p \in area(O) \\ c_1 \frac{\rho_O}{\eta} & , p \notin area(O) \end{cases}$$

For each object O which is projected on $area(O)$, ρ_O denotes the distance from p to the internal point $interia(O)$ and η the maximal distance from $interia(O)$ to the image boundary. This guarantees that the potential is within the interval $[0,1]$. The constants c_i are some scaling factors. To prevent labels from overlapping the object's silhouette, a repulsive force U_{silh} is defined:

$$U_{silh}(p) = \begin{cases} c_2 & , \rho_{silh} \leq \rho_S \\ 0 & , \rho_{silh} > \rho_S \end{cases}$$

where ρ_{silh} denotes the minimal distance from p to object boundary and ρ_S is the silhouette influence factor. Another repulsive force should prevent labels floating outside the image boundary:

$$U_{wall}(p) = \begin{cases} c_3(1 - \frac{\rho_{wall}}{\rho_W}) & , \rho_{wall} \leq \rho_W \\ 0 & , \rho_{wall} > \rho_W \end{cases}$$

where ρ_{wall} denotes the minimal distance from p to the image boundary and ρ_W is the boundary influence factor. Finally, the repulsive force of non-associated objects is considered by:

$$U_{rep}(p) = \begin{cases} c_4 & , p \in area(O_i) \wedge O_i \neq O \\ 0 & , else \end{cases}$$

The potential field for a given object O is defined as the sum of the attractive and the maximal repulsive force:

$$U(p) = U_{attr}(p) + max(U_{wall}(p), U_{silh}(p), U_{rep}(p))$$

Figure 3 depicts the potential field for the tibia, which is partly occluded by the ligamentum patellae.

3.4 Initial Label Placement

After establishing the static potential field, label particles move from their start position until they reach minima within the potential field. Currently, we experiment with a set of initial label positions: (a) the corners of projection, (b) the internal point of the largest object segment $\mathit{interia}(O)$, and (c) a point on the preferred main axis.

Preferred Main Directions: Human illustrators prefer to connect label and anchor point with horizontal or vertical lines (Criterion 3). This observation guides the estimation of preferred label placements. Penalty values for four vectors starting from the anchor pointing to the main directions are computed. The penalty function considers:

- the number of non-reference objects which the connecting line crosses,
- the length of segments, where the connecting line crosses non-reference objects,
- the length of segments, where the connecting line crosses the reference object,
- the amount of available free length to place the label and its distance from the anchor point.

The penalty function aims at minimizing the number of non-referring pictorial elements crossed by connecting lines and the distance between the anchor point and the label. Figure 4 presents the preferred main axis to connect reference object and label.

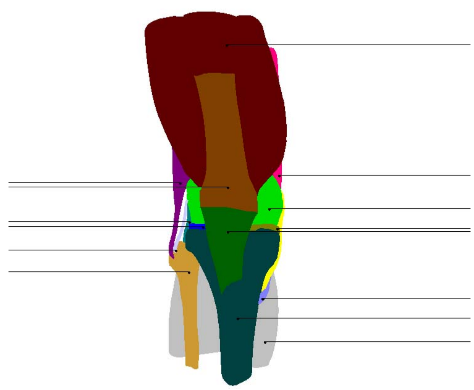

Fig. 4. Anchor points and preferred label directions.

3.5 Label Competition

While some of the label particles frequently reach identical local minima, there remain typically 3 different candidate positions. These candidates are measured according to a set of evaluation criteria: their *accumulated area potential* V_{pot}, their *visibility* V_{vis} (area not shared with other labels), the *length* V_{len} of the connecting line, and the minimal *angle* V_{angle} between the connecting line and either the horizontal or vertical axis. Due to space restrictions we will only present the first evaluation criterion:

$$V_{pot}(L) = \sum_{p \in \mathit{area}(L)} U(p), \text{ L refers to the label under consideration}$$

In order to compare these different measures, they are normalized into a standard range [0,1]. Therefore, for each pictorial element and each criterion the best and worst values of the label candidate set are determined.[2] We found good results using a weighted sum of these measures, which enables us to consider global layout considerations. Figure 5-Left presents the label configuration after completing the first phase of the FLOATING LABELS algorithm.

3.6 Dynamic Force Field Calculation

The evaluation criteria aim at an even label distribution over the available space. Due to the ease of label replacement overlaps are not completely prohibited. A greedy re-adjustment algorithm is based on the assumption, that it is more easy to determine alternative appealing label positions for large objects than those for tiny objects.

Therefore, label overlaps are determined iteratively. One of these labels is selected and pinned on its current position. The selection criteria consider the visible area and the area of an axis-aligned bounding rectangle. Pinned label establishes repulsive forces over an extended label area, which is added to the static potential fields of all objects except it associated object. Then positions for the remaining unpinned labels are re-adjusted according to the corrected static potential fields (Force E). Figure 5-Right presents the final label configuration. The FLOATING LABELS system also enables the user to apply manual corrections, while the system computes a balanced label configuration for the remaining labels.[3]

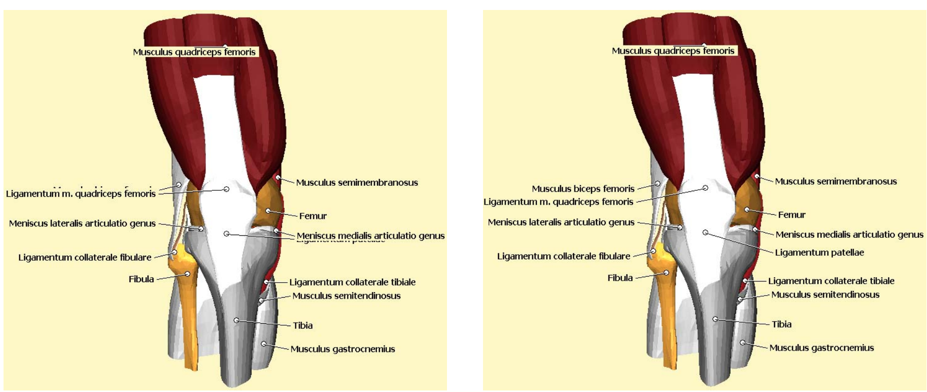

Fig. 5. Left: The initial label configuration for the knee joint. The model contains 19 different objects. 15 are visible from the current point of view, and 14 of them are big enough to place anchor points. **Right:** An improved label configuration. The greedy reposition algorithm was able to resolve both label overlappings.

---

[2] Note that these measures are either absolute (visibility) or relative (accumulated area potential, line length and angle).

[3] Not yet implemented.

3.7 Decoration

In this final step the meta-graphical objects (anchor points, connecting lines and labels) have to be integrated into an illustration in an aesthetic way. Therefore, we analyzed our corpus to extract appropriate render parameters for anchor points and connecting lines (e.g., shape, color). We found different rending styles:

Anchor Point Style: Some illustrators place them while others don't.

Connecting Line Style: Illustrators use either dashed or solid lines. Solid lines seem to segment the image into convex regions. In colored illustrations the choice of an appropriate line color is very difficult. Most illustrators prefer black, but they frequently use white when the connecting line overlays large dark regions. Frequently the connecting lines are absent for short distances between anchor point and label. Turns in the connecting line may also be used to prevent passing through dense, very detailed regions or to prevent label overlaps.

In order to guarantee the visibility in all regions, the current implementation uses solid white line with black shadows for both anchor points and connecting lines.

3.8 Software

The FLOATING LABELS algorithm is based on 2D projections of 3D geometric models from manually selected view points. Whenever hand-made illustrations of a similar object were available, we adjusted these view points of the 3D models to match these hand-made illustrations. Frequently, anatomic model are displayed from a set of different canonical viewing directions and thus several views of the same object are contained in our test set.

The 3D renderer is implemented in Open Inventor (Coin3D) and uses Qt to display the 2D image processing results. The basic algorithms are fully implemented, whereas the code and optimizations for user interactions have to be finished until the time of presentation. The runtime to compute the label layout for complicated views was up to 10 seconds on a modern PC.

4 Results

We tested our algorithm on more than 300 different geometric models taken from the ViewPoint library and Princeton 3D model search engine. Most of the geometric models do not separate individual objects. Even large 3D models with more than 100K vertices frequently did not segment them at all. For high-quality models the number of individual objects ranged between 10 and 25. The lack of highly segmented models was one of the main obstacles during testing.

Figure 6 presents some label configurations computed by the FLOATING LABELS system. Our algorithm fulfills Criteria 1–3 and 5. According to Criterion 4, crossings between connecting lines should be prevented as they are both distracting and misleading, the Criterion 6 refers to dynamic environments. Both criteria are not yet considered and subject to future work.

Within this limitation, we found the results quite promising. The label placement at local minima of potential field ensures an even label distribution. For illustrations with few labels even the label placement according to preferred main axis achieved good results. The consideration of repulsive force between labels in the second phase improves the label configuration on “hot spots”, i.e., regions where many labels have to be arranged within small spatial areas. However, the most challenging example proves, that the static as well as the dynamic force field have to be fine-tuned and adjusted.

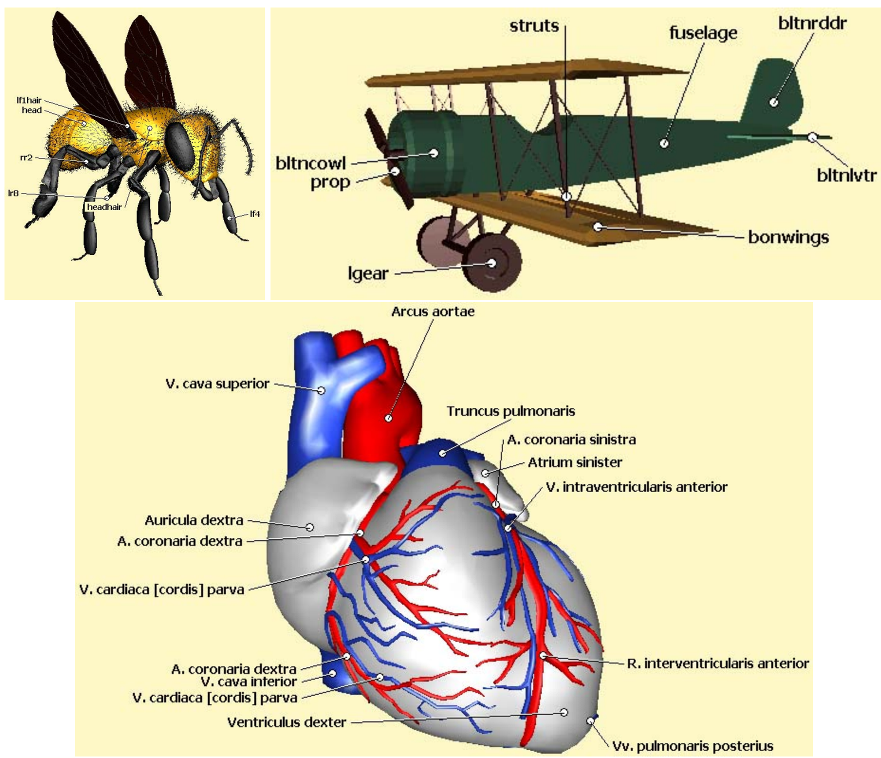

Fig. 6. Labels configurations automatically determined by the FLOATING LABELS system. For the heart model, the domain expert provided the denotation of geometric elements, whereas for the other models the internal descriptions are displayed in the label.

5 Improvement / Future Work

Human illustrators employ a variety of decoration styles (see Section 3.7). Hence, additional styles (e.g., labels without connecting lines, abbreviated label and legends) have to be integrated into the FLOATING LABELS system.

The current algorithm did not consider intersections of connecting lines. Based upon our observation how humans prevent unwanted intersections, we developed a strategy to resolve them. Whenever an intersection between two segments arises, the exchange of their label positions eliminates this intersection. Because labels do not overlap, this strategy can also be applied to labels of different sizes. A recursive application is able to resolve all intersections.

The main challenge is to integrate our force-field approach into an interactive 3D information system. While our first experiments use full-scaled projections, the reduction of quantization would speed-up the calculation considerably. Therefore, further experiments will analyze the accuracy-efficiency tradeoff. Moreover, hot spots can be observed after the initial label configuration, which might reduce the number of different potential fields required to balance them. Finally, alternative strategies to resolve conflicts, could be applied on hot spot areas (remove labels, shrink label text, alter height/width aspect of text boxes, legend creation . . .).

We plan to exploit shape approximations with bounding objects (object-oriented bounding boxes, sphere trees) to compute the forces and gradients analytically. This will reduce the amount of space required to store pixel-based approximations of potential fields and speed-up the calculation.

While the textual annotation or labels, as well as figure captions interrelate distant pictorial and textual segments, text balloon in comics embed almost all textual information within illustrations. Cartoon-style illustrations are especially attractive, as our algorithm can be also applied to determine appealing layouts for complex-shaped balloons. Special repulsive functions aim at avoiding visually important regions, like human faces and hands. Moreover, the algorithm is suitable to determine very complex layouts as required to design pictorial dictionaries. In all these application domains NPR shading techniques are most appropriate [18]. Hence, we will integrate our system into the OPENNPAR renderer [8].

6 Related Work

The automatic design of well balanced layouts is a central aspect within computer graphics, computational geometry, and cartography. A variety of different techniques have been applied to generate complex layouts automatically: constraints [11], simulated annealing [6], or genetic algorithms [12]. The authors have chosen the potential field method primarily as the observations presented in Section 1 could be transformed in a simple and elegant way to a variety of attractive and repulsive forces.

We created test set of several high-quality 3D models, and comparable hand-made illustrations, in order to extract human layout strategies and to evaluate our results. In order to compare different label layout techniques in terms of efficiency and visual balanced results, the authors plan to apply them on this test set.

The potential field method was successfully applied to plan paths for real objects like robots [10] as well as for virtual objects (e.g., camera planning in computer graphics). Several techniques were applied to prevent the object under control from getting stuck into local minima. To guide the planning, BECKHAUS [1] introduced navigation objects and dynamic potential fields. Our method has also to be compared with multi-agents path planning approaches in dynamic environments.

The appealing placement of labels is one of the central research subjects in cartography. Here, point-feature abstraction of labels have been frequently used (see [4]). EBNER [6] exploited force fields for the label number maximation problem for point features. Their approach aims at placing the maximal number of axis-aligned labels containing their reference points without any overlapping. In order to find global minima they combined a gradient-based and a simulated annealing approach.

PREIM [15] pioneered the interactive exploration of complex spatial configurations by visual and textual means. Textual annotation provides additional information about their co-referential geometric objects. Their content depends on the interaction context, which requires dynamic changes in size. Therefore, the spatial configuration of labels is adjusted by applying a 2D distortion technique. Moreover, PREIM and RAAB [14] employed mesh reduction to determine (multiple) anchor points for topographic complex geometric objects in 3D. However, the labels in these interactive information systems are either placed on special spatial area, placed manually or employ transparency to reduce the effect of occlusions.

BELL & FEINER [2] developed an algorithm to compute both covered and empty regions in dynamic environments, which was successfully applied for a dynamic label layout [3]. However, this algorithm is based on axis-aligned bounding boxes and does not yield best result for complex-shaped geometric object, whereas our algorithm works within the image space without such restrictions.

7 Conclusion

This paper introduces a label layout algorithm for complex-shaped geometric objects. It works on the image space without any approximation by bounding objects. Moreover, the label configuration is sensitive to global layout constraints, and can integrate user manipulations and automatic adaptations. Even though the FLOATING LABELS system is not yet integrated in an interactive 3D information system, our first experiments reveal a great potential of improvement and enhancement to speed-up calculations.

References

1. S. Beckhaus, F. Ritter, and T. Strothotte. Guided Exploration with Dynamic Potential Fields: The CubicalPath System. *Computer Graphics Forum*, 20(4):201–210, Dec. 2001.
2. B. Bell and S. Feiner. Dynamic Space Management for User Interfaces. In *Proc. of the 13th Annual Symposium on User Interface Software and Technology (UIST-00)*, pages 238–248. ACM Press, New York, 2000.
3. B. Bell, S. Feiner, and T. Höllerer. Information at a Glance. *IEEE Computer Graphics and Applications*, 22(4):6–9, July/August 2002.
4. J. Christensen, J. Marks, and S. Shieber. An Empirical Study of Algorithms for Point-Feature Label Placement. *ACM Transactions on Graphics*, 14:203–232, 1995.
5. *The Oxford-Duden Pictorial English Dictionary*. Oxford University Press, 1995.
6. D. Ebner, G. W. Klau, and R. Weiskircher. Force-Based Label Number Maximation. Technical Report TR-186-1-03-02, Institut für Computergraphik und Algorithmen, Technische Universität Wien, June 2003.

7. H. Gray. *Anatomy of the Human Body*. Lea & Febiger, Philadelphia, 20th edition, 1918. http://www.bartleby.om/107/.
8. N. Halper, T. Isenberg, F. Ritter, B. Freudenberg, O. Meruvia, S. Schlechtweg, and T. Strothotte. OpenNPAR: A System for Developing, Programming, and Designing Non-Photorealistic Animation and Rendering. In J. Rokne, R. Klein, and W. Wang, editors, *Proc. of Pacific Graphics 2003*, pages 424–428. IEEE Press, Los Alamitos, 2003.
9. O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.
10. J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.
11. S. Lok and S. Feiner. A Survey of Automated Layout Techniques for Information Presentations. In A. Butz, A. Krüger, P. Olivier, and M. Zhou, editors, *Proc. of the 1st International Symposium on Smart Graphics*, 2001.
12. P. Olivier, N. Halper, J. Pickering, and P. Luna. Visual Composition as Optimisation. In *AISB Symposium on AI and Creativity in Entertainment and Visual Art*, pages 22–30, 1999.
13. T. Pavlidis. A Thinning Algorithm for Discrete Binary Images. *Computer Graphics and Image Processing*, 13:142–157, 1980.
14. B. Preim and A. Raab. Annotation topographisch komplizierter 3D-Modelle. In P. Lorenz and B. Preim, editors, *Simulation und Visualisierung ’98*, pages 128–140. SCS-Society for Computer Simulation Int., Delft, 1998.
15. B. Preim, A. Raab, and T. Strothotte. Coherent Zooming of Illustrations with 3D-Graphics and Text. In W. A. Davis, M. Mantei, and R. V. Klassen, editors, *Proc. of Graphics Interface ’97*, pages 105–113. Canadian Information Processing Society, 1997.
16. F. Ritter, H. Sonnet, K. Hartmann, and T. Strothotte. Illustrative Shadows: Integrating 3D and 2D Information Displays. In W. L. Johnson, E. André, and J. Domingue, editors, *Proc. of 2003 Int. Conf. on Intelligent User Interfaces (IUI’03)*, pages 166–173. ACM Press, New York, 2003.
17. J. Sobotta, R. Putz, and R. Pabst, editors. *Sobotta: Atlas of Human Anatomy*. Lippincott Williams & Wilkins, Baltimure, 13. edition, 2001.
18. T. Strothotte and S. Schlechtweg. *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*. Morgan Kaufman Publisher, Los Altos, 2002.