Retrieval of technical drawings in DXF format - concepts and problems

Nadine Weber, Andreas Henrich
University of Bamberg
Chair of Media Informatics
D-96052, Bamberg, Germany
{nadine.weber, andreas.henrich}@wiai.uni-bamberg.de

Abstract

Nowadays, launching new products in short intervals is a critical factor for success to persist on the global market. At the same time many enterprises call for cost reduction in all of their divisions. Especially development departments have to increase their effectiveness and efficiency by lowering development time and costs to preserve the competitiveness of the company. Since development processes are affected by a plethora of information and knowledge, a starting point for time and cost reduction is to reuse existing design knowledge. This knowledge consists of both text documents and special kinds of artifacts such as 3D models, technical specifications, technical drawings, or bills of material. While the retrieval of text documents and 3D models is already explored widely in the research area, retrieving technical drawings is still a complex and interesting field of investigation. Therefore, this paper deals with existing search techniques for technical drawings and the problems emerging from implementing such techniques for *Drawing Interchange Format* (DXF) drawings. Moreover, we propose a general procedure for the extraction of features from such drawings to solve these problems.

1 Introduction

Today, the development of new products or product versions in the domain of mechanical engineering takes place by designing 3D models on modern Computer-Aided Design (CAD) systems. By automatically generating a drawing from a designed 3D model, these systems facilitate the creation of technical drawings needed for the manufacturing process. This leads to thousands of digital drawings stored in Product Data Management systems or on file systems. Thus, a tremendous source of information is available which is not further used in many cases; most of the data lies idle because their existence is unknown or they are not retrievable. Consequently, the reuse of such useful knowledge can contribute to reduce time and costs in the product development process.

For this purpose, a retrieval system is necessary to enable searching for technical drawings. But searching should not be seen as searching by name or drawing number only; it is the content of technical drawings that has to be addressed. Hence, both text-based retrieval methods (like, e.g., the vector space model or the boolean model) and content-based methods have to be applied to find appropriate results for a designer with a specific information need.

The first required step is to analyze the existing file formats. Since many different CAD systems are available, there is also a multitude of proprietary file formats which are not manageable without expensive converters. The only format with open access that can be handled by almost every CAD system is Autodesk's DXF format. DXF is a vector file format, expanded to a quasi-standard exchange format for technical drawings amongst nearly all companies. As a consequence, it makes sense to support this format with preference. Although the specification of this format is open and it is widely used, some aspects make implementing search techniques for it difficult.

For that reason, we want to consider some existing search techniques for technical drawings and their applicability for DXF drawings. Therefore, the paper is organized as follows. Section 2 examines our definition of a technical drawing. To this end, the potential information content of such a drawing is considered more precisely. In section 3 we give an overview of state-of-the-art search techniques for technical drawings in general. Before we sum up the paper by discussing our conclusions and presenting directions for further research in section 6, we point out the problems emerging from implementing a similarity concept for DXF drawings (cf. section 4) and thereupon make a proposal for how a feature extraction process can be configured (cf. section 5).

2 Definition of a technical drawing

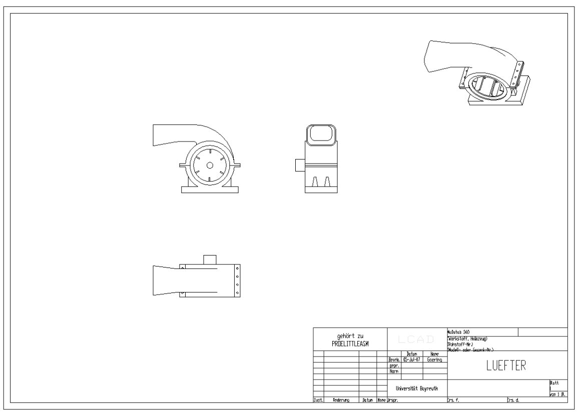

Figure 1: Simple example drawing of an exhauster.

For a better understanding of the following sections, we want to highlight our understanding of technical drawings first. According to [Conrad, 2005], we define a technical drawing as a line-based representation of scale of a part or assembly which consists of different views, slices and other additional information. Since technical drawings are an important means of communication between the design and manufacturing step in the product development process, it is necessary to consider them in searching for relevant information.

Therefore, we have to take into account that there are different types of drawings dependent on content, purpose, kind of representation, or type of creation. In [DIN, 1962] a diversity of terms such as sketch, plan, original, parent drawing, or preliminary drawing is defined. Considering these occurrences shows that a differentiation between a general drawing and a single part drawing is essential.

A single part drawing, as depicted in figure 1, predominantly describes the geometry of an individual part by displaying different views and slices of the product normally completed by dimensions. Additionally, such a drawing contains information about the product designation, the material, or tolerances integrated in form of one or more text fields. Moreover, bills of material or alternatives for the product can be contained. [Eigner and Maier, 1991]

General drawings in contrast are representations of assemblies consisting of multiple parts. The illustration of an assembly presents its mounting state to demonstrate the arrangement of the associated parts with their interdependencies. This kind of drawing is also denoted as exploded drawing or exploded view. As illustrated in figure 2, the parts are ordered in the way they would disrupt if the assembly detonated. [Vajna *et al.*, 1994]

Given the fact that we want to support engineering designers with relevant information, considering the information content of these drawings is an important task. General drawings provide mainly the structure of an assembly which can be used to create a topological representation of the product. Since this is the only information that can be extracted from such an information source, a precise study of the information content of a single part drawing has to be carried out. [Conrad, 2005] divides the content of a single part drawing into three categories: geometry data, technological information, and organizational information. While the former one gives a complete and detailed description of the geometry of primitive elements of the product such as lines, circles, or splines, the technological information contains, for example, dimensions, information about used material, or quality features. Furthermore, there is organizational information that can be divided into two groups. While the first group includes factual details such as designation and part numbers for identification and classification of the product, the second group comprises information referring to the drawing like scale, drawing format, charge number, draftsman, or creation date. Consequently, a search based on metadata as well as on geometry / topology is possible. Obviously, this requires different kinds of similarity concepts. For that reason, we give an overview of existing retrieval methods for engineering drawings in the next section.

3 Retrieval concepts for technical drawings

In general, technical drawings are illustrations of a designed product which can be stored in pixel or vector formats. Thus, a differentiation between pixel-based and vector-based retrieval methods has to be made. Obviously, both types need corresponding representations for

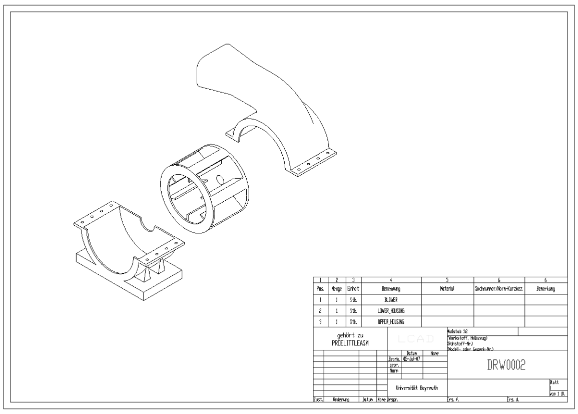

Figure 2: Exploded view of an exhauster.

the documents in the knowledge base and the query (the information need). The latter one can be posed in form of keywords, an example document (Query-by-Example QbE), or a sketch. While vector-based methods are rather suitable for keyword- and QbE-queries, pixel-based concepts are useful for sketches in form of scanned technical drawings or example documents.

3.1 Pixel-based methods

The following paragraphs give a review of existing retrieval concepts developed for image (pixel)-based engineering drawings.

Applying the Hough transform to extract global line features from a drawing is part of a retrieval method proposed by Fränti et al. [2000]. Thereby, the authors assume that engineering drawings are binary (black-and-white) images mainly consisting of line segments. For that reason, the process of line detection starts with the determination of the set of black pixels in the image. Then, each pixel is transformed into a parametric curve in the parameter domain which is also called the accumulator space. In doing so, each pixel (x, y) is described by means of the line equation $d = x \cdot \cos\theta + y \cdot \sin\theta$, where d is the distance from the origin to the line, and θ is the angle between the x-axis and the line's normal. Dependant on these two parameters, an accumulator matrix can be computed where each row corresponds to one value of the distance d and each column to one value of the angle θ. Thus, every pixel is arranged in this matrix, before a feature vector is generated. For this purpose, the authors suggest two variants. The first method reduces the matrix using a threshold value and sums up the significant coefficients in each column of the accumulator matrix. Hence, a global description of the image based on angular information is given. Since the use of this kind of information is not sufficient for large and more complex drawings, the authors present a second variant which includes positional information of the lines. This variant uses the full accumulator matrix for the generation of the feature vector and therefore allows more accurate image matching. The matching itself is done by applying a distance measure to the feature vectors of the query image and the database image. In [2001], Tabbone et al. present a method for indexing technical drawings based on the notion of F-signatures. In this approach, every binary graphical object in a drawing is represented by such an F-signature which is defined as a specific kind of histogram of forces. This histogram is generated by calculating all the forces exerted between the pixels of a same object. Therefore, a

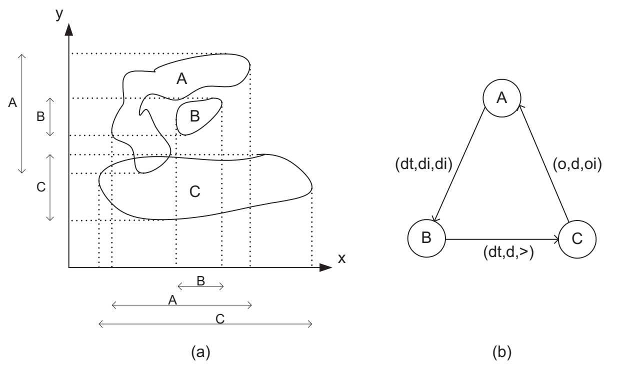

Figure 3: 2D-PIR of an example picture according to [Nabil *et al.*, 1996].

mapping function is used that is defined as $\varphi_r(d) = 1/d^r$, with d denoting the distance between two points (pixels) of an object. Dependant on the parameter r, different kinds of forces can be determined. For $r = 1$ for example, the attraction force between two points a_1 and a_2 is defined as $\varphi(a_1 - a_2)$. Thus, calculating the forces between all pairs of pixels of an object results in a force histogram describing the object. The matching between two objects is done by computing a similarity ratio of the two associated F-signatures. Although this kind of representation is characterized by low time complexity and invariance to scaling, translation, symmetry and rotation, it was developed for recognizing special kinds of graphical objects. Hence, this approach is rather suitable for characterizing technical drawings of the architectural domain by identifying objects such as a shower or a washbasin.

Another image-based approach was developed by Nabil et al., supporting the retrieval process by generating a representation called 2D Projection Interval Relationship (2D-PIR) [Nabil *et al.*, 1996]. Based on the 2D-string representation of Chang et al. [1987], a 2D-PIR is a symbolic representation of directional as well as topological relationships among spatial objects in a picture. In general, this concept adapts three existing representation formalisms, namely Allen's temporal intervals [Allen, 1983], 2D-strings, and topological relationships, and combines them in a novel way. As a result, a connected labeled graph is constructed, with nodes representing the objects of a picture in form of symbols (e.g. names) and edges illustrating the positional relationships between them (cf. figure 3(b)). Thereby, a positional relationship is described in form of a triple consisting of one topological and two interval relationships. While the topological relationship describes the correlation between the positions of two objects, the interval relationship constitutes a 'temporal' relationship between the objects. Therefore, an object is projected along the x and y axes resulting in an x-interval and an y-interval. On this basis, two objects are compared with regard to their 'temporal' appearance in the picture having, for example, a 'before', 'during', 'start', 'finish', or 'after' relationship. Figure 3 gives an example for a drawing with its corresponding graph consisting of three objects A, B, and C. The triple $(dt, d, >)$, for example, represents the 2D-PIR of the two objects B and C. The first parameter dt describes that the objects are 'disjoint'; the second parameter d illustrates that with regard to the objects' interval on the x-axis, object B appears 'during' object C. Finally, the third parameter $>$ specifies that the y-interval of object B lies 'after' the y-interval of object C. In this way, 2D-PIRs between all spatial objects in a picture are determined and a digraph is computed. The comparison of such two graphs consists in solving the graph isomorphism problem by applying similarity metrics both for topological as well as interval relationships.

Müller and Rigoll [1999] also present an approach for the description of image-based engineering drawings. Based on the use of stochastic models they represent a drawing image with a pseudo 2-D Hidden Markov Model (P2DHMM) which is surrounded by filler states. Thereby, a P2DHMM is defined as a stochastic automata with a two-dimensional arrangement of the states where states in horizontal direction are denoted as superstates. Moreover, each superstate is defined as a one-dimensional Hidden Markov Model in vertical direction. Since the generation of this kind of representation depends on a learning phase in which the P2DHMM is trained from specific graphic objects, this approach serves mainly for recognizing the learned objects in a drawing. As a consequence, applying this (pattern recognition) method for the retrieval of engineering drawings in real companies is not feasible because there is a huge amount of predefined objects that would have to be trained. For that reason, the concept of Müller and Rigoll is not further contemplated.

3.2 Vector-based approaches

Considering the pixel-based methods described in subsection 3.1 demonstrates that they are unsuitable for vector-based drawings because too much information gets lost or is not taken into account. Accordingly, retrieval approaches explicitly addressing the content of a technical drawing are needed.

The potential information content of a drawing as discussed in section 2 illustrates that searching for technical drawings should be based on both text and geometry / topology data. A method that tries to take into account both data types is described in [Love and Barton, 2004]. It is based on GT coding and was integrated in a commercial retrieval system called CADFind[1] supporting the CAD systems AutoCAD and SolidWorks. In this system, each drawing is represented automatically by a GT code. GT is the abbreviation for Group Technology and means that a part's geometry, material, and production process information is encoded into a string of digits or alphanumeric characters. Since an engineering drawing normally consists of several simple views of a part, the drawing is separated into these views at first. Afterwards, each view is extracted free of additions like title blocks, dimensions, or textual comments and serves as input to a feature extraction program. Given the resulting features, a GT code is generated. Since the authors did not explain their coding process in detail, a conventional coding and classification scheme such as the Opitz scheme [Opitz, 1970] can be used to form the basis of such a procedure. The Opitz classification scheme characterizes a part with respect to predefined properties (CAD features) of the part. For example, a part described by the partial code '01312' has the following geometrical characteristics:

- '0' = Rotational part with L/D $<$ 0.5 (first digit describes the component class)
- '1' = Stepped to one end, no shape elements (second digit specifies the external shape)

[1] www.sketchandsearch.com

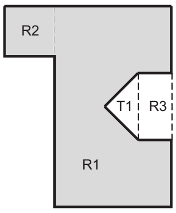

Figure 4: Example block according to [Park and Um, 1999].

- '3' = Smooth or stepped to one end with functional groove (third digit describes the internal shape)
- '1' = External planar surface (fourth digit gives information about plane surface machining)
- '2' = Axial holes related by a drilling pattern, no gear teeth (fifth digit specifies auxiliary holes and gear teeth).

Besides this geometrical description, the coding process includes also other additional information about the product such as material or production process information. Once the GT code is determined, both the code and an image of the view are stored in a record. Finally, two drawings (views) are supposed to be similar if they have similar GT codes and with it similar properties. However, implementing a retrieval system based on automatic coding and classification of drawings is no trivial task.

Except for this approach, normal text-based retrieval methods mainly provide no satisfying results. Consequently, the engagement in developing content-based retrieval concepts based on geometry and/or topology especially for technical drawings becomes a relatively new field of interest. For about ten years researchers from different countries have dealt with this subject.

Park and Um [1999] propose a method for content-based retrieval of technical drawings based on so-called dominant shapes. Here, the authors describe the contour of a complex graphic object by recursively decomposing its shape into dominant and auxiliary shapes. Moreover, they take into account topological information of the drawing by distinguishing between two types of spatial relationships: inclusion and adjacency. For this purpose, the authors remove dimension lines and characters from the drawing in the first step. Then, they partition the drawing into a set of dominant blocks, i.e. outstanding polygons formed along consecutive line segments describing, e.g. the views of a product. Furthermore, each block is separated into a set of shapes that can contain both polygons (blocks) and predefined primitives such as rectangles or circles. Thus, the description of a block results from adding or subtracting auxiliary shapes from a dominant shape. Figure 4 illustrates a simple example block consisting of a rectangle $R1$ (the dominant shape) extended by a rectangle $R2$ and reduced by the union of triangle $T1$ and rectangle $R3$. In this way, every arbitrary shape of a complex object can be described. Finally, the blocks are organized into a graph structure according to their inclusion and adjacency relationships to each other. The similarity between two drawings results from solving the graph matching problem. Furthermore, the recursive procedure applied in this approach enables partial matching of drawings, i.e. parts of a query drawing contained in other drawings can also be found.

Another approach, developed by Fonseca et al., draws on and expands the idea of Park and Um to improve the retrieval of technical drawings. In [Fonseca *et al.*, 2005], the authors perform two steps for the comparison of those artifacts. These steps consist in generating two representations which are used afterwards for similarity measurement. First of all, a representation based on topology information according to Park and Um's method described in the previous paragraph is created. Hence, a drawing is partitioned into blocks by isolating polygons. These polygons are described by their spatial relationships to each other and are represented in form of a topology graph. Since graph matching as used in [Park and Um, 1999] is an NP-complete problem, Fonseca et al. use graph spectra instead to solve the matching problem. As a result, for each topology graph a descriptor is computed by determining the eigenvalues of the graph's adjacency matrix. These values are stored in a multidimensional vector whose dimension depends on the complexity of the graph. Consequently, very complex drawings will result in vectors with high dimensions, while simple drawings will yield rather low dimensions. For that reason, the topological representations of the drawings are finally stored in an indexing structure called NB-tree which supports indexing of vectors with variable dimensions [Fonseca and Jorge, 2003]. By computing the euclidean norm a multidimensional vector is mapped to a 1D line and inserted in a B^+-tree. Moreover, a drawing is described on the basis of its contained geometry information. Fonseca et al. give two possibilities for extracting this data. On the one hand, a general shape recognition library called CALI, also developed by the authors, can be used. On the other hand, a computation of geometric features such as area and perimeter ratios from polygons such as the convex hull, the largest area triangle inscribed in the convex hull, or the smallest area enclosing rectangle amongst others is implementable. Applying the latter method to each polygon of a block gives a complete description of the block's geometry. On the basis of these representations, the matching procedure proves to be as follows. Searching the k nearest neighbors that have similar topological descriptors works as a filtering step to narrow down the result set. Afterwards, the geometrical information is used to refine the remaining drawings. The advantage of this method arises from the fact that a multi-level searching approach is possible. Generating different topology graphs for various levels of detail of the drawing provides searching for complete drawings as well as for subparts of these. However, it has to be taken into account that using graph spectra does not ensure the uniqueness of topology descriptors. Thus, more than one graph can have the same spectrum and with it the same descriptor.

Extracting shape appearances for the retrieval of engineering drawings is also used by Liu et al. [2004]. This method represents a drawing by an attributed graph where a

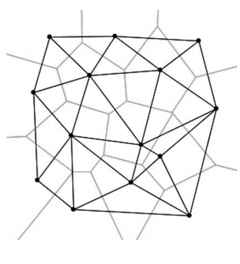

Figure 5: Voronoi diagram (gray) with corresponding Delaunay graph (black) (cf. http://de.wikipedia.org/wiki/Bild:Voronoi_delaunay.jpg, access: 2007-07-04).

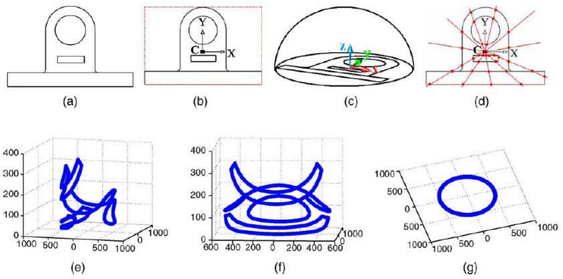

Figure 6: 2.5D spherical harmonics representation of a 2D drawing according to [Pu and Ramani, 2006].

node corresponds to a meaningful primitive extracted from the original drawing such as line or curve. Furthermore, for characterizing the content of an engineering drawing graph attributes are used which are divided into node attributes and edge attributes. While the former ones depict the appearance of the primitives such as circular, straight, or angular, the latter ones define the spatial relationships between these primitives such as parallel, or intersectant. The graph construction consists of four steps. First, each primitive is evenly sampled into multiple points which are adopted as input to a Delaunay tesselation in the second step. Thereby, a Delaunay graph is calculated which is defined as the dual graph of the Voronoi diagram of the set of points. Hence, after building the Voronoi diagram the graph can be constructed as follows: if two cells of the Voronoi diagram share an edge, the points located in the cells are connected. Figure 5 depicts both a Voronoi diagram (marked in gray) and the corresponding Delaunay graph (drawn in black). Afterwards, the resulting graph is simplified by merging nodes sampled from the same primitive into one node. Finally, as fourth step, the graph attributes are determined. By carrying out a Fourier transform with a direction histogram that describes the appearance of a primitive, the resulting coefficients of this transform are used as node attribute. On the other hand, the edge attribute contains elements such as the relative angle, the relative length, or the relative distance between two primitives. For graph matching the authors propose the application of the mean field theory which measures the similarity by calculating both the costs for matching graph edges and the costs for matching graph nodes.

Pu and Ramani deal with the problem of 2D drawing retrieval, too. In [Pu and Ramani, 2006], the authors propose two options, namely 2.5D spherical harmonics and 2D shape histogram, to find similar drawings for a query object as illustrated in figure 6(a). The first method draws on the successful application of the spherical harmonics representation in 3D shape matching. Thus, a drawing is described as a spherical function in terms of the amount of energy it contains at different frequencies. Therefore, the authors define a sphere whose center corresponds to the center of the drawing's bounding box and whose radius ensures to enclose the drawing completely (depicted in figure 6(b) and (c)). After that, a set of rays starting from the sphere center and locating in the plane where the 2D drawing lies, is generated (figure 6(d)). Determining the intersection points between these rays and the drawing serves as input to define a spherical function which is transformed from the 2D space into the 3D space. Finally, to compare two of these representations, a rotation-invariant descriptor is calculated by applying a fast spherical harmonics transformation method (figures 6(e)-(g) show the representation from different perspectives). The second approach of 2D shape histograms is a statistics-based representation originally developed by Osada et al. [2002]. For this purpose, a drawing composed of basic geometrical entities is transformed into a set of line segments. Afterwards, random points on these line segments are generated uniformly. The more points are sampled the more accurate is the approximation of the original shape. Once there are enough random points, the euclidean distance of every possible pair of randomly selected points is calculated. This distance is inserted into a histogram describing the distance distribution for the drawing. In the end, to measure the similarity between two histograms, the Minkowski distance is used.

Furthermore, technical line drawings can also be indexed by semantic networks. Yaner and Goel present in [2002] a retrieval process consisting of the two stages *reminding* and *selection*. In the first step, every drawing is represented by a feature vector, i.e. a vector of attribute-value pairs. Since drawings consist of different object types such as lines, circles, or ellipses, a feature vector is simply defined as a mapping from object type to its appearance frequency in the drawing. Consequently, a drawing is assumed to be similar to a query drawing if its feature vector is a superset of the query's feature vector. Given the results of this reminding step, the authors refine them by taking into account the spatial structure of the drawing. Therefore, the arrangement of the various object types in the drawing is described by five relation types called 'left-of', 'right-of', 'above', 'below', and 'contains'. To represent this spatial structure, the authors use a semantic network with nodes defining the spatial elements and links illustrating the spatial relations along them. On this basis, similarity between two drawings is determined in terms of subgraph isomorphism using symbolic methods; i.e. if the semantic network of the query can be found in the semantic network of a stored drawing, the latter one is delivered as similar.

4 Problems of implementing search techniques for DXF drawings

The concepts examined in section 3 prove that there are efforts to improve the retrieval of engineering drawings. However, implementing such methods for drawings especially based on the widely used vector file format DXF raises some problems that have to be solved. For that reason, the following paragraphs identify some of the problems a programmer is confronted with when processing data of a real technical drawing as illustrated in figure 7.

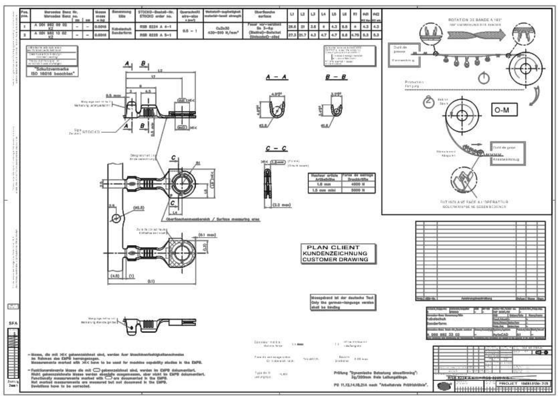

Figure 7: Example of a real technical drawing.

```
Reading HEADER...
    §ACADVER=AC1015
    §DWGCODEPAGE=ANS_1252
Reading CLASSES…
Reading TABLES…
Reading BLOCKS…
Reading ENTITIES…
Reading OBJECTS…
End of file.
=== 104088-0124_0125_C.dxf ===
    ==========
    ARC         :  143
    CIRCLE      :  6
    DIMENSION   :  29
    ELLIPSE     :  44
    HATCH       :  2
    INSERT      :  45
    LEADER      :  7
    LINE        :  288
    LWPOLYLINE  :  44
    MTEXT       :  7
    SPLINE      :  9
    TEXT        :  8
    Gesamt      :  632
    ==========
    DXF-Entitytyp : LWPOLYLINE
    Start line : 270364
    Entity reference : 7628
    Layer : 0
    DXF-Linetyp : BYLAYER
    DXF-Colorindex : BYLAYER
    Space : Model space
    Visibility : VISIBLE
    Scale of linetyp : 1.0
```

```
Typ : Closed Line
Nr of Points : 4
Position Point 1 : [572.9465,
273.0905]
Position Point 2: [831.0, 273.0905]
Position Point 3: [831.0, 586.0]
Position Point 4: [572.9465, 586.0]
============
DXF-Entitytyp : INSERT
Start line : 270398
Entity reference : 7629
Layer : 0
DXF-Linetyp : BYLAYER
DXF-Colorindex : BYLAYER
Space : Model space
Visibility : VISIBLE
Scale of linetyp : 1.0
BLOCK-Name : A$C7F4D0C4E
Insert position : [583.0277,
576.7610, 0.0]
X-Scale : 1.5
Y-Scale : 1.5
Z-Scale : 1.5
Angle : 0.0
Nr of Lines : 1
Nr of Columns : 1
==========
ATTRIB   :  2
BLOCK    :  1
Gesamt   :  3
==========
    DXF-Entitytyp : BLOCK
    Start line : 174376
    Entity reference : 3FC8
```

```
Layer : 0
DXF-Linetyp : BYLAYER
DXF-Colorindex : BYLAYER
Space : Model space
Visibility : VISIBLE
Blockname : A$C7F4D0C4E
External reference :
Reference Point : [0.0, 0.0, 0.0]
Switch : 2
    ==========
    ARC         :  102
    ATTDEF      :  2
    CIRCLE      :  18
    HATCH       :  1
    INSERT      :  3
    LINE        :  359
    LWPOLYLINE  :  176
    POLYLINE    :  47
    SPLINE      :  4
    TEXT        :  17
    Gesamt      :  729
    ==========
    DXF-Entitytyp : ARC
    Start line : 174402
    Entity reference : 3FC9
    Layer : TRAITFIN
    DXF-Linetyp : BYLAYER
    DXF-Colorindex : BYLAYER
    Space : Model space
    Visibility : VISIBLE
    Scale of linetyp : 1.0
    Center : [99.4575, -102.2113, 0.0]
    Radius : 41.4218
    Start angle : 184.2479
    End angle : 215.2192
    ============
```

```
DXF-Entitytyp : ARC
Start line : 174428
Entity reference : 3FCA
Layer : TRAITFIN
DXF-Linetyp : BYLAYER
DXF-Colorindex : BYLAYER
Space : Model space
Visibility : VISIBLE
Scale of linetyp : 1.0
Center : [98.0802, -103.6033, 0.0]
Radius : 40.2402
Start angle : 178.0175
End angle : 220.3958
============
DXF-Entitytyp : TEXT
Start line : 174454
Entity reference : 3FCB
Layer : COTES
DXF-Linetyp : BYLAYER
DXF-Colorindex : BYLAYER
Space : Model space
Visibility : VISIBLE
Scale of linetyp : 1.0
Text : Outil de pose
Position : [105.6937, -135.9725, 0.0]
Text hight : 1.0
Aspect ratio : 178.0175
STYLE name : ITALIC8
Horizontal alignment : Left
Vertical alignment : Baseline
============
DXF-Entitytyp : LINE
Start line : 174480
Entity reference : 3FCC
Layer : TRAITFIN
```

Figure 8: Part of the extracted content of the example drawing in figure 7.

This drawing contains a lot of useful information. On the one hand, the geometry of the product is shown from three views: one top view and two side views, depicted on the left side of the figure. In addition, slices according to the views are described, labeled as A-A, B-B, and C-C. Both views and slices are specified by dimensions. On the other hand, the drawing includes an amount of text information in form of text fields such as product designation, part number(s), material, project information or vendor data. Moreover, the right side of the drawing illustrates manufacturing information.

4.1 Missing file structure

Due to the mentioned information content above, the retrieval of technical drawings based on metadata is necessary. Thus, a designer should be able to search for a drawing by querying, for example, the part number or the project in which the drawing was created. Therefore, the available metadata has to be extracted. One main problem in doing so is the missing structure of a DXF file, which makes the extraction process quite difficult to accomplish. Since DXF was developed mainly as output format for plotters and as communication medium between designers and manufacturers, it is only suited for presentation. In correspondence to [Rudolph, 2000], examining a DXF drawing in more detail shows that it is a container of arbitrary objects with no interrelationships. In general, the objects are divided into two groups. The first group contains objects without any graphical embodiment such as dimensions, layers, line types, text types, or viewports. However, the second group, also referred to as entities, comprises objects with a graphical embodiment such as lines, circles, ellipses, polylines, splines, or blocks. The latter one combines a set of arbitrary objects into one object which can be used several times in a drawing. According to figure 8, a DXF file stores these objects in an arbitrary order, describing them mainly by their object type, their position on the drawing, and their geometrical data.

To extract metadata from a drawing, objects of type TEXT have to be identified. Although every textual information is stored in such an object, the understanding of its semantics is not ensured. Studying a text field on a drawing shows that it is composed of an abundance of LINE and TEXT objects. While the LINE objects possess no relevant information for metadata-based retrieval, we have to select all TEXT objects of a DXF file from which the textual data can be extracted. However, this procedure delivers the text, but not its meaning. Consequently, a part number, e.g. ’A2A00476’, can be extracted from the associated TEXT object. But there is no information that this text defines the part number. Instead, this information (the meaning) is also stored as text ’Part Number’ in a separate TEXT object which has no relationship to the number’s TEXT object. As a result, metadata extraction is possible, but taking into account the semantics of the extracted data is not supported. A possible solution for this problem could be the inclusion of the spatial proximity of the objects. Since objects that belong together, like ’Part Number’ and ’A2A00476’, are normally positioned close to each other on a drawing, finding adjacent TEXT objects (nearest neighbors) could help to find the matching ones. However, it has to be taken into account that there are often more than two objects in short distance to each other in a text field. Thus, identifying the right ones is a further challenge.

Furthermore, most of the concepts described in section 3 act on simple assumptions by using drawings consisting of only one block (a block defines a view of the product). Thus, these methods deal with technical drawings not comparable to real practical DXF drawings which contain normally more than one view. For example, Love and Barton define in [2004] a drawing as a simple view of a part that defines its essential geometry without any of the additions such as title blocks, dimensions, or textual comments, which are necessarily present on a normal engineering drawing. Hence, the different views of a real technical drawing have to be identified and separated. In doing this, the same problem of missing structure occurs. Although DXF provides so-called LAYER or VIEWPORT objects for defining several views of a product, this option is often not used by designers. All information – especially the view’s geometrical data – is mostly stored together in one LAYER object, hindering the identification of views. Moreover, this LAYER object contains the entities in an arbitrary order, i.e. the programmer has to find out which objects belong to which view. For example, a line can either be part of the geometry description of a view, or a dimensioning line, or it can be a boundary line of a text field. While in the first case the line is important for finding similar products based on geometry, a dimensioning line and a boundary line can be neglected for both a metadata and a geometry-based search. Hence, applying an appropriate algorithm for segmenting the drawing is needed which identifies the different views of the drawing together with their associated objects. Considering positional information can also be helpful for this task. However, identifying views with associated objects is a necessary step to be able to handle real technical drawings.

4.2 Different drawing layouts and format versions

Once the different views of a drawing are identified and separated, a further problem has to be solved. In general, the process of retrieving artifacts is based on posing a query. This can be done by either using a text describing the information need or by using an example object/document as input. The latter query-by-example option leads to the fact that all views of the drawings have to be compared. Thus, the occurring problem is that views used for a drawing are chosen variably from user to user, i.e. there is no standardization for views on a drawing. While one designer represents a product, for example, with a top view and two side views as in figure 7, another designer displays the same product with a front view, only one side view, and a back view. This leads to the fact that it is not ensured that all drawings have the same views. Thus, the emerging question is what views should be compared or, rather, how significant is the similarity measured between two views for the whole document. Consequently, retrieving technical drawings requires a form of indexing, and with it a filtering step that takes into account the kind of views contained in the drawing.

Another problem in handling technical DXF drawings arises from the fact that designers configure their drawings in different ways. Although the data which has to be on a drawing is predefined in most cases, there is no unique layout for the drawings. In general, the ISO norm 5457 [ISO, 1999] defines the sizes and the general layout of technical drawings, but, dependent on suppliers and customers, a company often has to adapt the layout to the specific guidelines of the supplier or customer. As a consequence, a company has to deal with a multitude of different drawing layouts. Besides, every designer has its own idea of presentation (i.e. there is no uniform use and notation of LAYER objects and no uniform placing of text blocks) what complicates working with this kind of CAD-specific artifact.

Finally, DXF is a format that is in constant development, i.e. with every new version of the CAD system AutoCAD Autodesk also provides an improved DXF version. Consequently, the retrieval of technical drawings has to take into account different format versions, and with it different kinds of objects dependent on the used version. While the older objects are well documented, the new ones are not. Thus, considering all possible objects of a DXF format is not feasible. For this reason, a restriction to the core objects according to [Rudolph, 2000] makes sense. Concentrating on objects which are of capital importance and which are most commonly used, such as lines or circles, is a first possible solution for this problem. Nonetheless, a lot of information gets lost by disregarding the new objects. This can lead to the fact that products are not described completely and a retrieval system might generate false results.

5 Feature Extraction Process

Solving the problems described in section 4 requires a feature extraction process consisting of two parallel paths. Consequently, we propose the procedure depicted in figure 9 to generate both a metadata and a geometry representation for a DXF drawing. One path of this process consists in extracting the metadata of a technical DXF drawing. For this, a *Metadata Extractor* is needed which selects all the TEXT objects from the DXF drawing. Afterwards, relationships between the TEXT objects have to be identified. This task is carried out by a *Metadata Correlator* which has to find metadata that belong together such as 'Designer' and 'John Q. Public'. Therefore, the *Metadata Correlator* has to implement methods that consider the spatial proximity of textual objects. However, it is also conceivable to include concepts from the domain of Optical Character Recognition (OCR). These methods enable the identification of single text objects and give the possibility of identifying their relationships and semantics. Since there are OCR approaches that take into account the context of a text object, a differentiation between typical text attributes, which are contained in every drawing (e.g. drawing number, creation date, or the designer's name), and other additional text information can be conducted.

The other path of the feature extraction process generates a geometry representation for a drawing. First, a *Layout Eliminator* identifies and eliminates all the elements that determine the layout of the drawing. Moreover, all dimensions contained in the drawing have to be rejected. To this end, a *Dimension Eliminator* is used. After these two operations, the drawing only contains the real geometrical information in form of product views and slices. Hence, the different views of the drawing have to be determined. This function is realized by a *Drawing Decomposer* which applies a segmentation method to partition the drawing.

Finally, the *Representation Generator* has to transform the extracted information (metadata and geometry) into suitable representations (e.g. feature vectors) and has to store them in index structures.

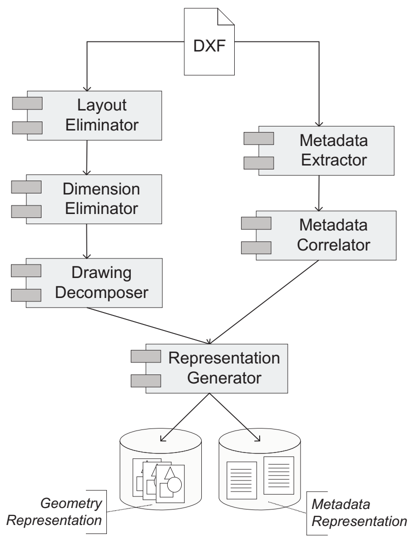

Figure 9: Proposed feature extraction process for DXF drawings.

6 Conclusion and Future Work

Retrieving technical drawings is an eminent help for engineering designers in doing their everyday work. Thus, a retrieval algorithm has to be implemented which considers both the textual information in form of metadata, and the geometrical data describing the geometry of the illustrated product. Since text-based retrieval methods are widely explored in research, our paper presents two groups of geometry / topology-based concepts: pixel-based and vector-based approaches. While the former do not take into account the real information content of a drawing and technical drawings are actually generated in vector formats, the latter group of similarity methods should be used for a retrieval system. However, there is a diversity of proprietary file formats (one for each available CAD system) that cannot be handled without having expensive converters. Since Autodesk's DXF format is the only open format in this domain, we decided to base our ideas for retrieving technical drawings on this file format. But using DXF implicates several problems as demonstrated in this paper. Thus, implementing a search functionality for technical drawings is no trivial task.

Therefore, we propose a procedure for extracting both metadata and geometry information from a DXF drawing. We started our implementation of this process by extracting the textual information of a drawing. We created a *Metadata Extractor* that simply selects all TEXT objects from the drawing. The results are indexed by a *Representation Generator* using an Apache Lucene[2] index. Hence, our future work has to concentrate on building the connections between the metadata and its meaning to improve the index. In addition to this task, we also dealt with content-based retrieval methods for DXF drawings. Concerning this matter, we implemented two algorithms from section 3. Currently, these algorithms work on simple drawings containing only one view of a product and no additional information. Thus, in our next working steps we have to evaluate the algorithms in regard to their suitability for practical use. Therefore, a concentration on identifying the different views of a real technical drawing by eliminating both layout elements and dimensions is necessary.

Acknowledgments

The work described in this paper is part of the Bavarian research alliance FORFLOW consisting of six Bavarian universities working in the fields of engineering design and computer science. It collaborates with 21 companies and is promoted by the Bavarian Research Foundation (BFS). [3]

References

[Allen, 1983] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November 1983.

[Chang *et al.*, 1987] S. K. Chang, Q. Y. Shi, and C. W. Yam. Iconic Indexing of 2D Strings. *IEEE Transactions of Pattern Analysis and Machinable Intelligence*, 9(3):413–428, 1987.

[Conrad, 2005] Klaus-Jörg Conrad. *Grundlagen der Konstruktionslehre: Methoden und Beispiele für den Maschinenbau*. 3. aktualisierte und erweiterte Auflage, Carl Hanser Verlag, München/Wien, 2005.

[DIN, 1962] Deutsches Institut für Normung. *DIN 199: Technische Zeichnungen - Benennungen*

[2]http://lucene.apache.org

[3]For more details see http://www.abayfor.de/forflow/en

[Eigner and Maier, 1991] M. Eigner and H. Maier. Einstieg in CAD: Lehrbuch für CAD-Anwender. 8. Nachdruck, Carl Hanser Verlag, München/Wien, 1991.

[Fonseca and Jorge, 2003] Manuel J. Fonseca and Joaquim A. Jorge. Towards Content-Based Retrieval of Technical Drawings through High-Dimensional Indexing. *Computers and Graphics*, 27(1):61–69(9), February 2003.

[Fonseca *et al.*, 2005] Manuel J. Fonseca, Alfredo Ferreira, and Joaquim A. Jorge. Content-Based Retrieval of Technical Drawings. *Int. Journal of Computer Applications in Technology*, 23(2-3):86–100, March 2005.

[Fränti *et al.*, 2000] Pasi Fränti, Alexey Mednonogov, Ville Kyrki, and Heikki Kälviäinen. Content-based matching of line-drawing images using the Hough transform. *Int. Journal on Document Analysis and Recognition*, 3(3):117–124, 2000.

[ISO, 1999] International Organization for Standardization. ISO 5457:1999, Technical product documentation - Sizes and layout of drawing sheets. 1999.

[Liu *et al.*, 2004] R. Liu, T. Baba, and D. Masumoto. Attributed Graph Matching Based Engineering Drawings Retrieval. In *Document Analysis Systems VI - Proc. of the 6th Int. Workshop*, pages 378–388, Florence, Italy, September 2004. Springer.

[Love and Barton, 2004] Doug Love and Jeff Barton. Aspects of design retrieval performance using automatic GT coding of 2D Engineering Drawings. *4th Int. Conf. on Integrated Design and Manufacture in Mechanical Engineering*, Bath, April 2004.

[Müller and Rigoll, 1999] Stefan Müller and Gerhard Rigoll. Searching an Engineering Drawing Database for User-specified Shapes. In *Proc. of the Fifth Int. Conf. on Document Analysis and Recognition*, pages 697–700, Washington, DC, USA, 1999. IEEE Computer Society.

[Nabil *et al.*, 1996] Mohammad Nabil, Anne H. H. Ngu, and John Shepherd. Picture Similarity Retrieval Using 2D Projection Interval Representation. *IEEE Transactions on Knowledge and Data Engineering*, 8(4):533–539, August 1996.

[Opitz, 1970] Herwart Opitz. A Classification System to Describe Workpieces. Pergamon Press, Oxford, 1970.

[Osada *et al.*, 2002] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape Distributions. *ACM Transactions on Graphics*, 21(4):807–832, October 2002.

[Park and Um, 1999] Jong H. Park and Bong S. Um. A new approach to similarity retrieval of 2-D graphic objects based on dominant shapes. *Pattern Recognition*, 20:591–616, 1999.

[Pu and Ramani, 2006] Jiantao Pu and Karthik Ramani. On visual similarity based 2D drawing retrieval. *Computer-Aided Design*, 38(3):249–259, March 2006.

[Rudolph, 2000] Dietmar Rudolph. AutoCAD-Objekte. SYBEX-Verlag GmbH, Düsseldorf, 2000.

[Tabbone *et al.*, 2001] S. Tabbone, L. Wendling, and K. Tombre. Indexing of Technical Line Drawings Based on F-Signatures. In *Proc. of the Sixth Int. Conf. on Document Analysis and Recognition*, pages 1220–1224, Los Alamitos, CA, USA, 2001. IEEE Computer Society.

[Vajna *et al.*, 1994] S. Vajna, Chr. Weber, J. Schlingensiepen, and D. Schlottmann. CAD/CAM für Ingenieure: Hardware - Software - Strategien. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 1994.

[Yaner and Goel, 2002] Patrick W. Yaner and Ashok K. Goel. Using Spatial Structure in the Associative Retrieval of 2-D Line Drawings. Technical Report GIT-CC-02-70, Artificial Intelligence Laboratory College of Computing, Georgia Institute of Technology, Atlanta, USA, December 2002.