This ICCV workshop paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

# 2D Cross-View Object Segmentation and Perceptual Grouping in Computer-Aided Design Drawings

Mohamed Dhia Elhak BESBES
Capgemini
mohamed-dhia-elhak.besbes@capgemini.com

Zahra VAHIDI FERDOUSI
Capgemini
zahra.vahidi-ferdousi@capgemini.com

Hedi TABIA
Univ Paris Saclay
hedi.tabia@univevry.com

Mouna FRADI
Capgemini
mouna.fradi@capgemini.com

## Abstract

*This paper introduces our methods in creating a comprehensive evaluation resource for assessing the capabilities of algorithms aimed at segmenting and perceptually grouping 2D mechanical technical drawings. Our dataset encompasses a diverse collection of such drawings, accompanied by semi-automated annotations of segments and groups. These annotations were reviewed by domain experts, following detailed guidelines to ensure both consistency and top-notch quality. The dataset is intended to serve as an invaluable asset for researchers dedicated to advancing techniques that enhance the comprehension and interpretation of 2D mechanical drawings.*

## 1. Introduction

Technical drawings serve as a universal language that enables engineers, designers, and manufacturers to effectively communicate intricate details and specifications of a product. By capturing essential dimensions, tolerances, and geometric configurations, technical drawings lay the foundation for successful and precise manufacturing processes.

One of the primary purposes of technical drawings is to convey critical tolerance information that ensures the accuracy and functionality of the end product. Tolerances define the permissible variations in dimensions, ensuring that the manufactured components fit together seamlessly and perform as intended. These drawings provide a visual representation of how different parts fit and interact within an assembly, aiding in the identification of potential issues and streamlining the production process. However, understanding and interpreting these drawings can be challenging due to the complexity and variability of their content. To address this problem, researchers have developed various techniques for automatically segmenting and grouping the elements in technical drawings, with the goal of improving their readability and comprehension. There has been a significant amount of research on techniques for segmenting and perceptually grouping elements in 2D mechanical technical drawings.

**Datasets** In research papers, most of the work done on technical drawings refers to the architecture, engineering and construction industries. We can cite [6], which presents a large-scale real-world CAD drawing dataset of floor plans. Similarly to vehicle conception drawings, architecture drawings are composed of line segments, arcs, curves, and texts.
The authors of [1] proposed a benchmark of four kinds of drawings: mechanical, architectural, and two distinct types of utility drawings.

On the other hand, segmentation datasets encompass a wide range of domains, offering extensive coverage. The authors of [22] propose a large dataset for semantic segmentation of medical images. The analysis of urban scenes is another application([20], [2]), used essentially for autonomous driving or urban security concerns. The authors of [12] propose a composite dataset from different domains (Cityscapes [2], Mapillary [16], COCO [13], ADE20K [27], and others), in order to do cross-domain semantic segmentation by training a single generic model that can be applied to images even not seen during training.
(Video object segmentation [18], instance segmentation [8].)
Multi-view segmentation in also applied to different fields and images, like plants [21], Jacquard patterns (in textil field) [24], radars [17] or car detection [9]. [19] and [11] exploit the geometry of 3D mesh reconstructed from multi-view images (e.g. for scene understanding).

Between the research done on 3D graphical models and images, we can also cite [14], where the authors prepared a dataset of computer-generated sketch data. The dataset is collected from CAD databases, and can be used for building a retrieval system for 3D CAD models. [10] proposes an approach for segmenting 3D shapes into labeled semantic parts. The key idea of their approach is to combine image-based fully convolutional networks for view-based reasoning, with a surface-based projection layer, for more coherent shape segmentations. [26] proposes a method to synthetically generate a large amount of technical drawings in order to be used for training drawing component segmentation models. Their method is essentially based on the randomization of the dimension sets under two constraint conditions. In addition to datasets specifically designed for 2D drawings, there are also datasets that provide segmentation annotations for 3D CAD models. While these datasets may not be tailored exclusively to 2D drawings, they hold the potential for inferring 2D segmentations. Among such datasets, noteworthy examples include: The study by [25] introduces a minimalist language encompassing fundamental sketching and extrusion modeling operations. Their approach is underpinned by a dataset containing a substantial collection of 8,625 human-designed sequences, all meticulously expressed within the confines of this simplified language. [4] introduces the CC3DOps dataset, featuring 37k CAD models annotated with operation type and step labels. Additionally, they present the CADOps-Net, a deep neural network that cohesively learns CAD operation types and their step-wise decomposition.

**Technical/engineering drawings analysis** The authors of [23] propose a hybrid approach to build a parser which automatically interpret technical drawings. They combine image segmentation (DBSCAN) and object recognition approaches (CNN, ResNet-50) for extracting and identifying the principal drawing components, and ILP approaches for extracting tables properties.
[5] proposes a novel technical drawings vectorization method, based on deep vectorization model and primitive optimization approach, with pre-processing (cleaning) and post-processing (minimizing the number of primitives) steps. [3] highlights the dual nature of engineering drawings (mixture of objects and annotations) and proposes a CAD conversion strategy based on the iterative nature of understanding engineering drawings by expert humans. The authors summarize its major functions as follows: primitive recognition, syntax-based annotation analysis and layer separation, functional analysis of each view, and 3D reconstruction.
The authors of [15] propose a general framework for engineering drawing digitization based on the following steps: drawing pre-processing, shape detection, feature extraction and representation, classification, and contextualization. While exploring the state-of-the-art approaches, they highlight two major challenges for applying CNNs in this domain: the lack of sufficient annotated examples, and the complexity for designing a generic digitization platform due to the diversity in term of image quality ranges, standards and rule sets for complex engineering drawings.
The authors of [7] focus on view and section identification of technical drawings. They model the relationship among various views by a graph representation and validate these relations by an approach based on evidence theory.

## 2. Dataset

### 2.1. Technical drawings

The technical drawing of a part is considered as a technical and graphical communication tool which aims at facilitating the interpretation of a particular shape. This drawing provides, in the form of a plan, the tolerances and dimensions required for manufacturing. This graphical language is based on a set of criteria and rules known as international standards and composed of the following components:

- the objects composing the part, represented by different types of lines, circles, ellipses and arcs.
- the different views of the part (front, back, left, right, top, bottom, transverse and perspective views),
- the title bloc known also as information block and located in the bottom right-hand corner of the page. This block provides a better understanding of the technical drawing and contains the part name, the designer's name, the drawing number, the measurement unit, the material of the construction, the scale, the angle of projection, etc.
- the objects quotations.

In this work, we focus on images, and ignore the textual parts of the technical drawings, except quotations.

As mentioned beforehand, the objects composing the part are represented through different types of lines. Each line has a specific signification and can be:

- continuous line: this type represents the physical boundaries of an object. We can find thick lines used on the outside contour and thin lines on the inner contour,
- hidden line: the dashed lines are drawn to represent invisible or hidden edges of the object behind the main lines,
- center line: drawn to show the centers of the round elements,

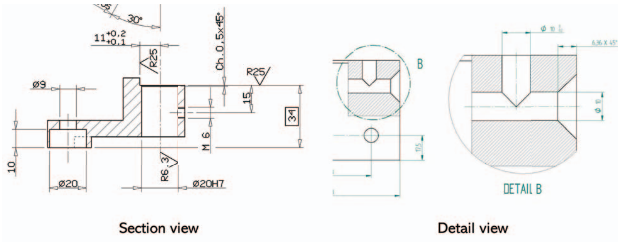


Figure 1. Example of a section and detail view

- section line: these lines are used to show the internal sections like holes in section planes,
- extension and dimension line : generally represented as thin lines and drawn to show dimensional features and lengths of the objects in technical drawings,
- cutting plane line: intermittent and thick lines, used to represent a plane in which sectional view is taken.

The different lines detailed above constitute the part's views in the technical drawing. Each view has a particular purpose and should not duplicate information already shown in another view. The most common views used in an engineering drawing are : front, top and side views. Nevertheless, some additional views can be necessary to represent more information such as section and detail view (see figure 1).
2D Technical drawings can offer information absent in 3D CADs or highlight certain aspect of 3D CADs. In both cases, 2D drawings tend to avoid redundancy i.e. information available in 3D CADs. Hence, the dataset that we present in this work is not generated from 3D CADs but rather manually constructed by experts.

## 2.2. Annotation

We collect a set of technical drawings (a total number of 371) used in vehicle design, by the aid of multiple mechanical experts. For confidentiality, we mask some sensitive information in their title blocs. The drawings are complex, and contain various entities like lines, circles, ellipses, arcs and quotations.
To annotate our "real life" technical drawings, we develop a graphical interface (by JAVA and the swing API), which let the user (here a mechanical expert) to (1) identify the different views, (2) specify the different objects in each view, and (3) to do the correspondence of objects among the different views. The resulting annotations will be saved in JSON format. In the following parts of this section, we describe each one of these functionalities.

### 2.2.1 View Detection

As said before, a technical drawing is composed of multiple elements, especially the different views of a mechanical part: front, back, left, right, top, bottom, perspective, cross sections, zoomed and combined views.
The user can identify each view by its type by a bounding box (Figure 2). The type of the view and the coordinates of the left upper and the right down points of the bounding box are saved in a JSON file for each technical drawing.

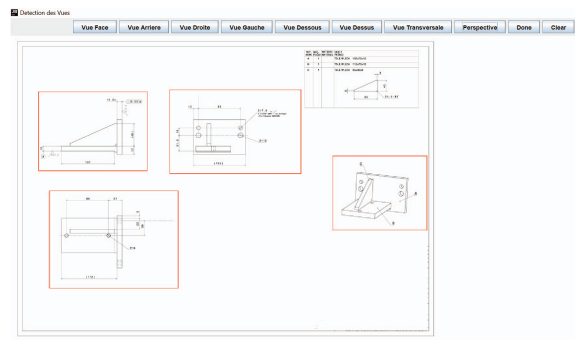


Figure 2. The view identification module of the graphical interface

### 2.2.2 Object Segmentation

After identifying the different views, the user will be redirect to the object segmentation module, where he can specify the different objects of each view using a list of graphical primitives. The list is composed of the followings elements: continuous line, dashed line, double dashed lines, circles, ellipses and arcs.
As showed in Figure 8, the user can specify each object by validating it via a "validate object" button. By this act, each object segmentation will be shown in a different color, and its corresponding annotation will be saved in a JSON file. The content of the JSON file will be discussed in section 2.3.

In this phase, the object annotations are done by the user clicks. The resulting images and their annotations will be used to train an image recognition algorithm, which will be tested on real technical drawings. So it is important to have images the most similar to the original ones. To guaranty the quality of the generated images and annotations, we propose to the user some functions to let him produce almost exactly the same images:

**Line:** The user can create geometric forms like square or rectangle by specifying their composed lines. To draw a line, we ask the user to click on its two endpoints.

As said before, the quality of the resulting image is primordial. To this end, we make available the following options: to guaranty the well alignment of vertical and horizontal lines, the user can specify it by pushing the "Shift" keyboard key. The Figure 3 illustrates the difference between two versions without using the alignment function (3.a), and by using it (3.b).

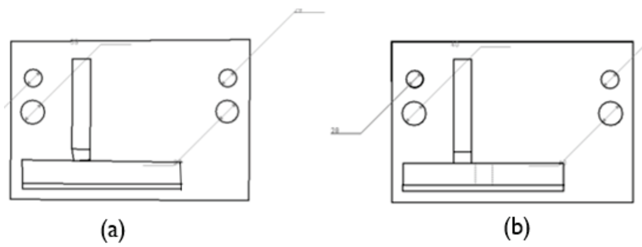


Figure 3. The "Shift" aligned closing point function result

Also, to draw a closed form, we allow the user to align his first and last clicks by pushing the "CTRL" keyboard key. This action allows us to have a well closed form. The Figure 6 illustrates the difference between the two versions without using this function (4), and by using it (5).

**Circle:** For drawing a circle, we ask the user to click on the upper and bottom points of the circle. Based on these clicks, we compute the x and y indexes of the upper left corner of the circle bounding box, and the diameter of the circle (equal to its width and height).

**Ellipse:** To reproduce an ellipse, we ask the user to click on its two longer extremity points and a shorter one. Based on these three points, we compute the center of the ellipse and its rotation angle, which are necessary to draw it.

To obtain a specific arc, we need to know four data points of it. So, for drawing an arc, we ask the user to specify the following data points (illustrated in Figure 7: the two endpoints of the arc, its middle extremity point, and one middle points between an endpoint and the middle extremity point.

The Equation 1 illustrates the equation of a conic. We consider an arc as part of an ellipse. So to find the arc points, we first solve the equation system 2, to find solutions for the coefficients $A, B, C$ and $D$. Then we consider the arc bounding box and explore the points in this bounding box, by going trow them, iteratively, line by line (and then column by column). In each iteration, we compute the Equation 1 of the points, and select the one with the nearest equation solution to one, as an arc point.

$$Ax^2 + By^2 + Cx + Dy = 1 \qquad (1)$$

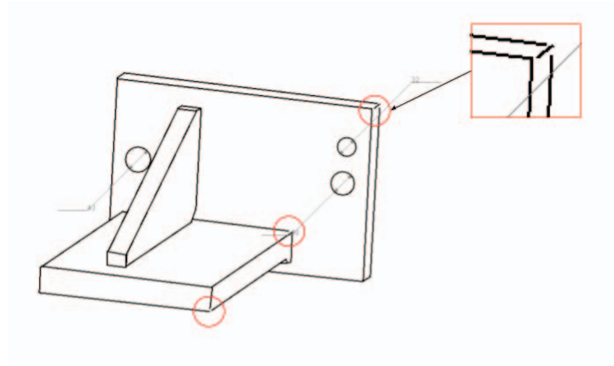

Figure 4. Without "Ctrl"

[b]0.3

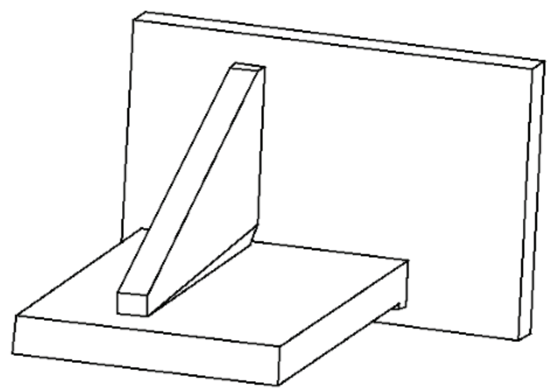

Figure 5. With "Ctrl"

Figure 6. The "CTRL" aligned line function result

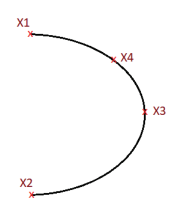


Figure 7. An example of the arc data points asked to the user

$$\begin{bmatrix} x_1^2 & y_1^2 & x_1 & y_1 \\ x_2^2 & y_2^2 & x_2 & y_2 \\ x_3^2 & y_3^2 & x_3 & y_3 \\ x_4^2 & y_4^2 & x_4 & y_4 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad (2)$$

To guarantee a correct segmentation, a particular attention must be paid by the mechanical expert, while specifying the different objects. In our case, each shape variation in the main part is considered as a new object and should be annotated using a different color. Furthermore, several

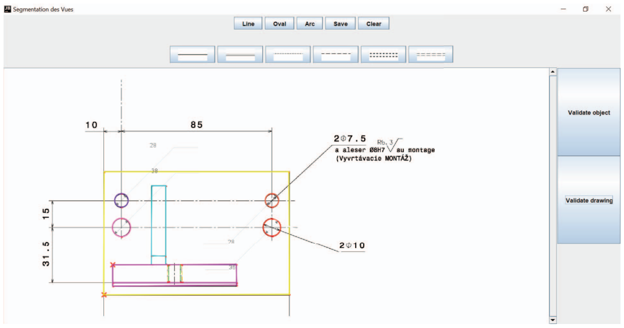


Figure 8. The object segmentation module of the graphical interface

parts in technical drawings are composed of an assembly of multiple sub-parts. Thus, we consider these sub parts as different objects that need to be annotated with distinct colors. An example of object segmentation is illustrated in figure 9. Due to the complexity of some technical drawings, an in-

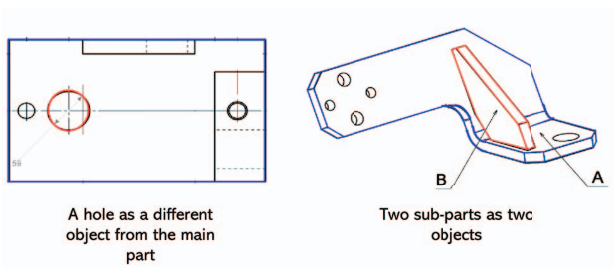


Figure 9. Object segmentation in a technical drawing from a mechanical expert viewpoint

appropriate segmentation can be done. The most common mistakes that can be made during the object specification are:

- the annotation of different objects using the same color: in this case the mechanical expert considers two distinct objects as a unique one. This can be happen when the difference between the objects is not clearly illustrated in the engineering drawing. An example of this unsuitable segmentation is shown in Figure 10. The dashed lines in this drawing represent two holes with different diameters which must be colored differently.

- the annotation of the same object with two different colors: this error can be made when an object is represented in a view with more than a unique geometric form. We consider the example shown in figure 11. The threaded hole is represented through an inner thick circle and an outer thin three-quarter of a dashed circle. These circles have to be annotated as the same object using the same color.

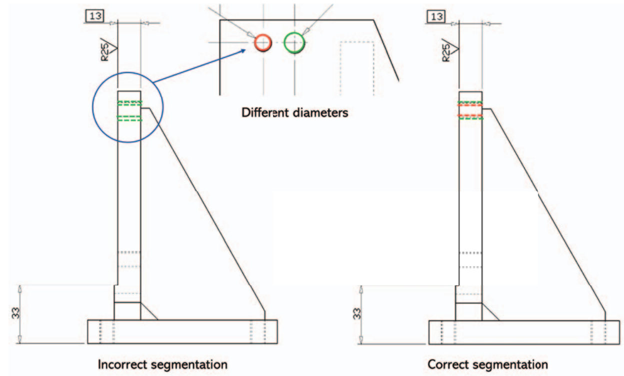


Figure 10. Annotation of two holes

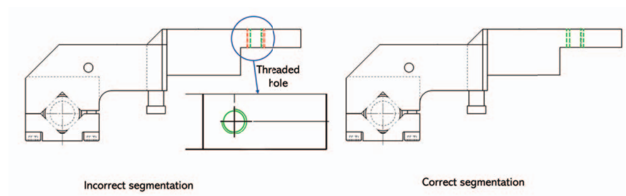


Figure 11. Annotation of a threaded hole

### 2.2.3 Object annotation

As mentioned before, the annotations of segmented objects of technical drawings are saved in a JSON file. In this annotation file, for each object, we have a list of its composed elements. The type of the element could be different types of lines, circles/ellipses or arcs. Depending on the element type, the following data will be annotate:

- *Line:* the coordinates of the two endpoints of the line,
- *Circle and Ellipse:* the coordinates of the upper left corner of the circle/ellipse bounding box, the circle/ellipse width and height, its rotation (0 for circles) and the coordinates of its center point.
- *Arc:* the coordinates of the upper left corner of the arc bounding box, its width and height and the list of arc points.

An example of an annotation file is illustrated in Figure 12, where we can find a part composed of two objects: the first one is composed of two continuous lines and an arc, and the second is an ellipse.

In order to train deep learning models on our data, we transform the annotations in the coco format [13]. In this format, we specify some information about each object: its area, bounding box, category id, image id, segmentation type (is crowd or not crowd) and segmentation list of vertices ($x$, $y$ pixel positions).
There is two different types of annotations: *is crowd* and

*not crowd*. The former consists in iterating on the image and giving each pixel a boolean value, whether it belongs or not to the object. Therefore, the pixels that belong to an object are colored as shown in the Figure 14, while the latter one is composed of segments that encapsulate the object as illustrated in Figure 15.

In this work, we found the *not crowd* annotation type more efficient, so to specify the segmentation of objects, we contour its composing elements by polygon vertices as follows (also illustrated in Figure 13):

- *Line:* each line is surrounded by a rectangle based on two parallel lines with a small margin.

- *Circle/Ellipse:* to segment a circle, we need an exterior circle linked to an interior one. To obtain the exterior circle, we set its $x1$, $y1$ (coordinates of the upper left corner of its bounding box) to $x1 - margin$ and $y1 - margin$, and a value of $2 * margin$ will be added to the width and height. Then for each pixel of this bigger bounding box, we compute the circle equation, and save the points belonging to it. A similar procedure will be conducted to obtain the interior circle (with a smaller bounding box, by $x1 + margin$, $y1 + margin$, $width - 2 * margin$ and $height - 2 * margin$). We keep one point of four of these two linked circles to construct the circle segmentation.
  The same procedure is used for ellipses based on the ellipse equation.

- *Arc:* to segment an arc, we specify an exterior segment linked to an interior one based on the coordinates of the 5 origin arc points.


```
[{
  "color": 1,
  "objects": [
  { "data": [ 202, 348, 202, 428 ],
    "type": "continueLine"
  },
  {
   "data": [ 373, 370, 373, 433 ],
   "type": "continueLine"
  },
  {
   "data": [ 202, 428, 171, 29 ],
   "origin": [ 202, 428, 373, 433, 290, 457, 241, 449, 338, 452, ··· ],
   "type": "arc",
   "points": [ 200, 431, 201, 431, 202, 428, 203, 428, 204]
  }]
 },
 {
  "color": 2,
  "objects": [
  { "data": [ 231, 330, 32, 12, 338, 244, 342],
    "origin": [ 259, 348, 229, 336, 242, 348],
    "type": "ellipse"
  }]
 }
]
```


Figure 12. An extract of the resulting JSON file for the annotation of a technical drawing view.

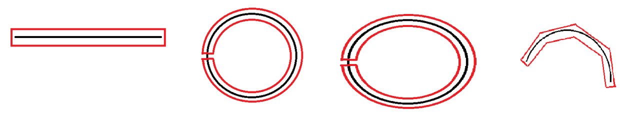

Figure 13. Segments of different forms

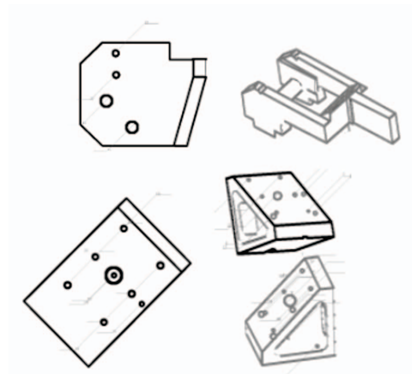

Figure 14. Samples of the crowded annotation type.

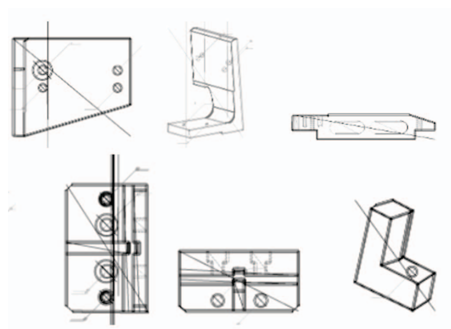

Figure 15. Samples of the not crowded annotation type.

#### 2.2.4 Perceptual Grouping

To guarantee a correct segmentation, it is essential to match the objects in different views which refer to the same entity. For this purpose, we developed a module in the graphical interface (Figure 16) that exposes the different views with the segmented objects in different colors. Via the colors palette, the user can define the correspondence of objects in different views. The module will save the correspondence annotations in a JSON file.

Based on the segmentation done previously, the object correspondence can be defined. This correspondence consists in matching an object with its correspondents represented in the different views using the colors palette. An example of an object correspondence in four different views is illustrated in figure 17.

It is important to note that any inappropriate segmentation can be the origin of some mistakes while defining

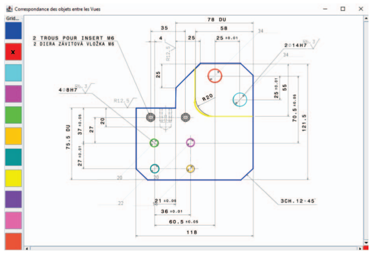

Figure 16. Object correspondence module of the graphical interface.

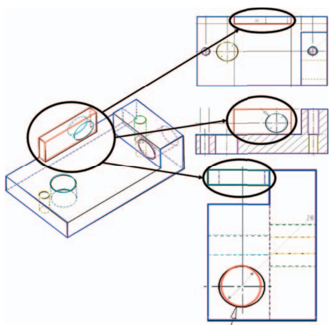

Figure 17. Object correspondence form a mechanical expert viewpoint

the correspondence. Therefore, the mechanical expert must verify that the objects specification is done correctly. However, even though the segmentation is done in a correct way, some confusions can be present during the objects correspondence. In diverse technical drawings, some superposed objects in a view can be found. Since the engineering drawings are based on the concept of orthographic projection, this superposition is always present. Nonetheless, the placement of one object above or on top of another one, makes the correspondence task difficult for the mechanical expert. Thus, to tackle this issue, the mechanical expert must do the correspondence of one object and ignore the second one. We present in Figure 18 an example of superposed objects in a technical drawing.

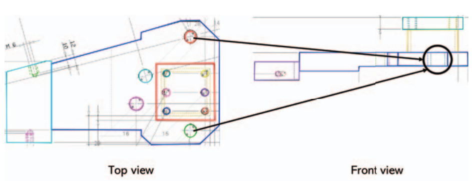


Figure 18. Two holes in the top view represented by one object in the front view

Furthermore, another case can be presented while corresponding the different objects, where we can find some objects represented in a view and hidden in another one. Figure 19 illustrates this case.

Moreover, the oval holes can be source of some confusions during the objects correspondence. In various technical drawings, the orthographic projection of these holes are represented by four dashed lines, two outer lines to represent the rounded corners of the oval hole and two inner ones for its axis. This representation is similar to an orthographic projection of two simple holes as shown in figure 20 which is confusing in some cases. For this purpose, the mechanical expert must verify the technical drawing to make sure that the drawing is well understood.

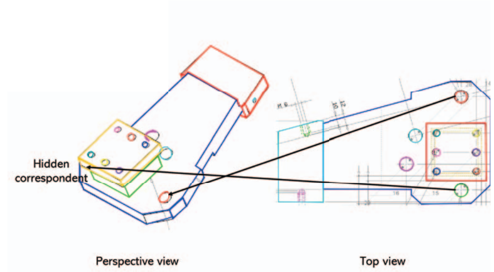


Figure 19. A hidden correspondent of a hole in the perspective view

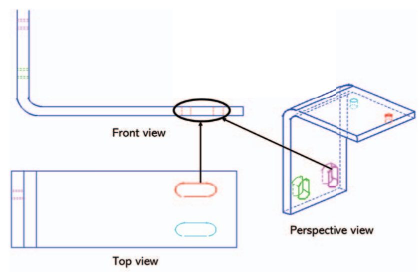


Figure 20. Example of the correspondent of an oval hole in a technical drawing

### 2.3. Resulting dataset characteristics

In this section, we describe the dataset of mechanical drawing segmentation and perceptual grouping that was used to evaluate the proposed approach and the baselines.

#### 2.3.1 Description

The dataset consists of 371 mechanical drawings with annotated segments and groupings, where 371 is the number of drawings in the dataset. The drawings are obtained from a variety of sources, including technical manuals, blueprints, and CAD models, and cover a wide range of styles and complexity levels. The segments are manually annotated by expert drafters and represent the different elements in the drawings, such as lines, symbols, and annotations. The groupings are also annotated by expert drafters and represent the perceptual organization of the segments into meaningful units, such as views, components, and subassemblies.

The dataset is divided into a training set, a validation set, and a test set, with a ratio of 60%, 20%, and 20%, respectively. The training set is used to train the proposed approach and the baselines, the validation set is used to tune the hyperparameters of the models, and the test set is used to evaluate the final performance of the models.

#### 2.3.2 Statistics

Table 1 shows the statistics of the dataset of mechanical drawing segmentation and perceptual grouping, including the number of drawings, segments, and groupings in the training set, validation set, and test set. The table also shows the average number of segments and groupings per drawing in each set.

The dataset is balanced, with a similar number of segments and groupings in each set. The average number of segments and groupings per drawing is 75, which represents a moderate level of complexity for the task of mechanical drawing segmentation and perceptual grouping. The dataset includes a diverse range of styles and complexity levels, which allows for the evaluation of the proposed approach and the baselines in a wide range of scenarios.

| Database count | | | |
|---|---|---|---|
| Drawings | Views | Objects | Groups |
| 371 | 825 | 7120 | 744 |

Table 1. Statistics of Technical Drawings Database: Number of Drawings, Views, Objects in a View, and Groups Across Views

| Vue type count | |
|---|---|
| Vue | Count |
| FRONT | 267 |
| TOP | 183 |
| LEFT | 86 |
| PERSPECTIVE | 334 |
| REAR | 3 |
| TRANSVERSE | 34 |
| RIGHT | 51 |
| BOTTOM | 34 |

Table 2. Statistics of view detection

## 3. Measurements

There are several ways to measure the visual quality of the groupings generated by an approach for mechanical drawing segmentation and perceptual grouping. Here are a few examples:

Qualitative assessment: This method involves having expert drafters assess the visual quality of the groupings generated by the approach and provide a subjective rating. The ratings can be collected using a structured questionnaire or a simple scale (e.g., from 1 to 5). The advantage of this method is that it captures the nuances of the groupings and allows for a more detailed analysis of the strengths and weaknesses of the approach. The disadvantage is that it is subjective and may be affected by the biases of the evaluators.

Objective metrics: This method involves using objective metrics to quantify the visual quality of the groupings generated by the approach. Some examples of objective metrics that can be used include:

Grouping error: This metric measures the average error between the groupings generated by the approach and the annotated groupings in the evaluation dataset, using a suitable distance metric. A lower grouping error indicates a higher visual quality.

Grouping compactness: This metric measures the compactness of the groupings generated by the approach, i.e., the extent to which the segments in a grouping are close to each other. A higher compactness indicates a higher visual quality.

Grouping legibility: This metric measures the legibility of the groupings generated by the approach, i.e., the extent to which the segments in a grouping are clearly separated from the segments in other groupings. A higher legibility indicates a higher visual quality.

The advantage of this method is that it allows for a more objective and systematic evaluation of the visual quality of the groupings. The disadvantage is that it may not capture all the aspects of the visual quality and may be affected by the choice of the metrics.

Ultimately, the choice of the measurement for the visual quality of the groupings will depend on the goals of the evaluation and the resources available. A combination of qualitative and objective measurements may provide a more comprehensive evaluation of the visual quality of the groupings.

## 4. Conclusion

In this paper, we presented a dataset of mechanical drawing segmentation and perceptual grouping for the evaluation of approaches for the automatic analysis of mechanical drawings. The dataset consists of 371 drawings with annotated segments and groupings, covering a wide range of styles and complexity levels. The dataset is divided into a training set and a test set, with a ratio of 80% and 20%, respectively. The dataset is balanced and includes a moderate level of complexity, with an average of 7120 segments and 744 groupings per drawing.

The dataset of mechanical drawing segmentation and perceptual grouping is a valuable resource for the research community, as it allows for the evaluation of approaches for the automatic analysis of mechanical drawings in a consistent and fair manner. We hope that the dataset will facilitate the development of new and improved approaches for this important task.

## References


[1] A Chhabra and I Phillips. A benchmark for graphics recognition systems. *Empirical Evaluation Techniques in Computer Vision, IEEE Comp Press, CA, USA*, 1998. 1

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 1

[3] Dov Dori and Karl Tombre. Paper drawings to 3-d cad: A proposed agenda. In *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pages 866–867. IEEE, 1993. 2

[4] Elona Dupont, Kseniya Cherenkova, Anis Kacem, Sk Aziz Ali, Ilya Arzhannikov, Gleb Gusev, and Djamila Aouada. Cadops-net: Jointly learning cad operation types and steps from boundary-representations, 2022. 2

[5] Vage Egiazarian, Oleg Voynov, Alexey Artemov, Denis Volkhonskiy, Aleksandr Safin, Maria Taktasheva, Denis Zorin, and Evgeny Burnaev. Deep vectorization of technical drawings. In *European Conference on Computer Vision*, pages 582–598. Springer, 2020. 2

[6] Zhiwen Fan, Lingjie Zhu, Honghua Li, Xiaohao Chen, Siyu Zhu, and Ping Tan. Floorplancad: A large-scale cad drawing dataset for panoptic symbol spotting. *arXiv preprint arXiv:2105.07147*, 2021. 1

[7] Jie-Hui Gong, Hui Zhang, Bin Jiang, and Jia-Guang Sun. Identification of sections from engineering drawings based on evidence theory. In *Proceedings of the 2008 ACM symposium on Solid and physical modeling*, pages 13–24, 2008. 2

[8] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019. 1

[9] Derek Hoiem, Carsten Rother, and John Winn. 3d layoutcrf for multi-view object class recognition and segmentation. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 1

[10] Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 3d shape segmentation with projective convolutional networks. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3779–3788, 2017. 2

[11] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3d semantic segmentation. In *European Conference on Computer Vision*, pages 518–535. Springer, 2020. 1

[12] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2879–2888, 2020. 1

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 5

[14] Bharadwaj Manda, Shubham Dhayarkar, Sai Mitheran, VK Viekash, and Ramanathan Muthuganapathy. 'cadsketchnet'-an annotated sketch dataset for 3d cad model retrieval with deep neural networks. *Computers & Graphics*, 99:100–113, 2021. 2

[15] Carlos Francisco Moreno-García, Eyad Elyan, and Chrisina Jayne. New trends on digitisation of complex engineering drawings. *Neural computing and applications*, 31(6):1695–1712, 2019. 2

[16] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017. 1

[17] Arthur Ouaknine, Alasdair Newson, Patrick Pérez, Florence Tupin, and Julien Rebut. Multi-view radar semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15671–15680, 2021. 1

[18] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016. 1

[19] Hayko Riemenschneider, András Bódis-Szomorú, Julien Weissenberg, and Luc Van Gool. Learning where to classify in multi-view semantic segmentation. In *European Conference on Computer Vision*, pages 516–532. Springer, 2014. 1

[20] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 1

[21] Weinan Shi, Rick van de Zedde, Huanyu Jiang, and Gert Kootstra. Plant-part segmentation using deep learning and multi-view vision. *Biosystems Engineering*, 187:81–95, 2019. 1

[22] Amber L Simpson, Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram Van Ginneken, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint arXiv:1902.09063*, 2019. 1

[23] Dries Van Daele, Nicholas Decleyre, Herman Dubois, and Wannes Meert. An automated engineering assistant: Learning parsers for technical drawings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15195–15203, 2021. 2

[24] Wenzhen Wang, Na Deng, Binjie Xin, Yiliang Wang, and Shuaigang Lu. Novel segmentation algorithm for jacquard patterns based on multi-view image fusion. *IET Image Processing*, 14(17):4563–4570, 2020. 1

[25] Karl D. D. Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G. Lambourne, Armando Solar-Lezama, and Wojciech Matusik. Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences, 2021. 2

[26] Wentai Zhang, Quan Chen, Can Koz, Liuyue Xie, Amit Regmi, Soji Yamakawa, Tomotake Furuhata, Kenji Shimada, and Levent Burak Kara. Data augmentation of engineering drawings for data-driven component segmentation. 2

[27] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 1