Chemical Structure Recognition: A Rule Based Approach

Noureddin M. Sadawi, Alan P. Sexton, and Volker Sorge
School of Computer Science, University of Birmingham, UK

ABSTRACT

In chemical literature much information is given in the form of diagrams depicting molecules. In order to access this information diagrams have to be recognised and translated into a processable format. We present an approach that models the principal recognition steps for molecule diagrams in a strictly rule based system, providing rules to identify the main components — atoms and bonds — as well as to resolve possible ambiguities. The result of the process is a translation into a graph representation that can be used for further processing. We show the effectiveness of our approach by describing its embedding into a full recognition system and present an experimental evaluation that demonstrates how our current implementation outperforms the leading open source system currently available.

Keywords: Chemical molecule recognition, diagram recognition

1. INTRODUCTION

Documents in the chemical and life sciences, such as journal articles, patents, etc., often include diagrams depicting molecules. Such diagrams efficiently communicate important information to knowledgeable human readers, but is in general inaccessible by electronic means. Making this information accessible would not only help readers with visual impairments but would make it visible to search tools, thereby enabling better retrieval and exploitation of important chemical knowledge. As a consequence a significant amount of work has been done in recent years on the recognition of molecule diagrams from images. A number of commercial (see Refs. 1–4) and academic tools (see Refs. 5–7) have been developed to support this task. In addition there have been approaches developed for recognition of handwritten chemical diagrams (see Refs. 8–12) as well as efforts to produce ground truth data for a wide variety of molecule depictions.[13] Most of the recognition approaches work in a procedural manner, with the recognition steps roughly definable as an initial vectorisation, recognition of the particular artefacts that occur in molecule diagrams, an OCR step often using some machine learning technique such as neural nets and, finally, the translation into a graph data structure that enables the easy generation of chemical representations in languages like SMILES[14] or MOL.[15] Thus the recognition process is usually based on a set of heuristics that need to be experimentally tuned. The only exception appears to be the chemOCR project,[6,7] which claims to apply a rule based approach within an expert system to reconstruct the molecule description from graph and character information of the molecule structure. However, from the literature it is unclear how this rule based approach works or what the single rules would be.

In this paper we present the basis for a strictly rule based system for the recognition problem of chemical molecules. The main idea is to define precise, overlap free rewrite rules that transform a vectorised molecule image into a representation that can be captured by a convenient data structure, such as a graph. And while our approach incorporates a number of parameters that enable possible variation and fine tuning, the rules are designed to resolve possible ambiguities in a purely declarative, heuristic-free manner. A first implementation of our approach has yielded promising results in comparison with the leading freely available recognition system OSRA.[5] We also achieved a recognition rate of 95% and 94.9% on two runs of 1000 diagrams as part of the Image2Structure task of the TREC 2011 Chemistry track.[16]

In § 2 we will first further motivate our problem before defining some of the preliminaries (§ 3) and the rule system (§ 4). We then briefly discuss our first implementation of the system (§ 5) and present some experimental results (§ 6), before concluding with suggestions for improvements.

2. EXAMPLE

Fig. 1 contains an image of a fairly standard molecule structure that can be found in the literature. One can observe that it essentially is a graph structure, where vertices are atoms or sub-molecules and the edges are different types of bonds. Vertices are either given explicitly — by names of atoms (e.g., O) or more complex molecules (e.g., HO) — or implicitly by corners of bonds representing omitted carbon atoms. Bonds can be of different types, for instance in Fig. 1 we have

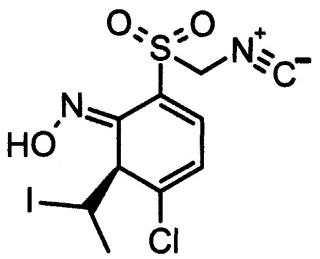

Figure 1. A Molecule Example

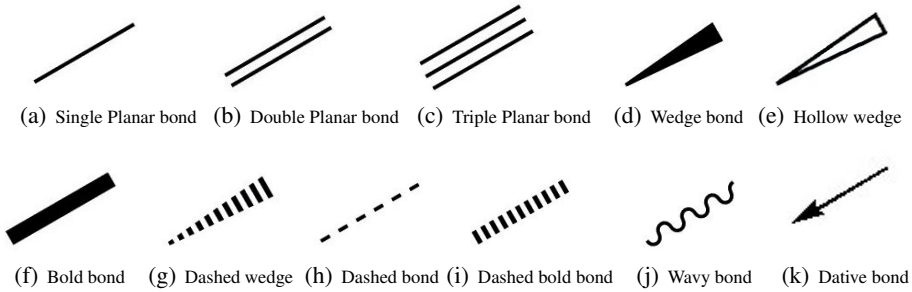

Figure 2. Common Bond Conventions

single, double, and triple bonds, indicated by one, two, or three lines between vertices but also so called stereo bonds, represented by a wedge, that indicate a direction in 3D space.

An overview of the different types of bonds is given in Fig. 2. Molecule diagrams often include elements that indicate their 3-dimensional structure. As the figure illustrates, a solid wedge, a hollow wedge or a bold line, Figs. 2(d), 2(e) and 2(f) respectively, are used to show bonds coming out of the plane of the drawing surface (up, toward the viewer). Each of these bond types has a direction to determine what chemists denote as the *stereo-centre*. The direction of the solid and hollow wedge bonds is determined by the tip-to-base direction, meaning the stereo-centre is at the narrow end. A dashed wedge, a dashed line and a dashed bold line, Figs. 2(g), 2(h) and 2(i) respectively, are used to depict bonds going into the plane of the drawing surface (down, away from the viewer). These also have directions to specify the stereo-centre. For a dashed wedge bond, the stereo-centre is at the shortest dash, so the direction is from the shortest to the longest dash. For both dashed bold and bold bond types, direction and stereo-centre are unspecified and cannot be determined by purely syntactic methods but have to be guessed heuristically taking context information into account. A wavy bond, as in Fig. 2(j), is used to indicate an unspecified configuration (mixture of up and down). Also, as Fig. 2(k) shows, an arrow is used to illustrate a dative (polar) bond. The direction of the arrow is from source-to-head and it indicates the existence of a negatively charged atom at the head of the arrow. Finally, one or more normal line(s) are used to show planar bond(s) as in Figs. 2(a),2(b),2(c).

One can easily observe a number of problems a recognition algorithm for molecule structures faces: For example, in Fig 1 the triple bond could be recognised as a dashed bold bond were it not for the presence of the atom names that give it direction. One can also not easily assume a minimum length for lines representing bonds, as for example the line between HO and N is much shorter than anything else in the image, but nevertheless represents a single planar bond. Other problems can occur when confusing characters with bonds as for example can happen with the capital letter "I", lower case letter "l", or the minus sign in Fig. 1.

3. PRELIMINARIES

We first define some of the concepts on which to base our rule system.

3.1 Geometrical Primitives

Our first assumption is that we work with a restricted set of geometrical primitives that are the result of an initial vectorisation phase of the image. We distinguish five primitives.

Line Segment: We define a line segment l as a rectangle with measurable width and length, a unique centre point $c(l)$, and two endpoints $e_1(l), e_2(l)$ representing the middle points on the edge of the short sides of the rectangle.

Arrow: A line segment with an arrow head as one end point.

Circle: A ring consisting of the circumference (of some unspecified thickness) of a circle. In molecule diagrams this is found only as the indicator of an aromatic ring inside a pentagonal or hexagonal ring.

Triangle: A solid wedge shaped with a discernible base line.

Character Group: A group of letters, digits or symbols. They are closely grouped spatially and are considered one entity.

Note that there is no "polyline" object as all connected paths of line segments are split into sets of simple line segments at their corners by our initial vectorisation. In particular, any intersecting set of line segments will be split at the junction point. Thus line segments in the form of a sans serif "T" will be split into 3 separate but connected line segments, and in the form of a sans serif "V" into 2 line segments.

3.2 Concepts

We need a number of general concepts to define our rules, based on parameters that can be instantiated later. We must allow both for digitisation artifacts in the images of the molecular diagrams and for imprecision in the drawing of the original images. Thus our rules must accommodate a certain level of approximation in attributes of components of the diagram such as slopes of lines, collinearity of sets of points, lengths of lines etc. We make use of *fuzzy parameters* for this purpose. These are pairs of values that record the minimum and maximum allowable values for their attribute of interest. For a fuzzy parameter $f = [f_{min}, f_{max}]$ and a value v, we say that v is approximately f, or $v \in f$, if and only if v is between the minimum and maximum values allowed by f.

Separation Distance: The distance between two graphic objects is defined as the closest distance between points of the different objects. For objects p and q, we abbreviate separation distance as $dist(p, q)$.

Connected: We say that two line segments are connected if they have exactly one endpoint in common.

Dash-Neighbouring: Let x be a fuzzy parameter. An element of a set of objects is said to *dash-neighbour* (wrt. x) another element of that set if the two elements are distinct and their separation distance is approximately x.

Approximate Collinearity: Let x be a strict (i.e. non-fuzzy) parameter. A set of points is defined to be *approximately collinear* with respect to *Radius of Collinearity* x, if there is a line such that the orthogonal distances from the line to each of the points is less than or equal to x. We say that a set of geometrical objects is approximately collinear if the aggregate set of all points of the objects is approximately collinear.

Approximately Parallel with separation x and overlap y, $\|_x^y$: Let x, y be strict parameters. Let l_1, l_2 be line segments. Then $l_1 \|_x^y l_2$ if and only if

(i) for the lines L_1, L_2 fixed by the end points of l_1, l_2, respectively, every point $p \in l_1$ has $dist(p, L_2) \leqslant x$ and every point $p \in l_2$ has $dist(p, L_1) \leqslant x$.

(ii) there exist line segments $l_1' \subseteq l_1$, $l_2' \subseteq l_2$ with $length(l_1') = length(l_2') = y$ and for every $p \in l_1'$ $dist(p, l_2') \leqslant x$, and for every $p \in l_2'$, $dist(p, l_1') \leqslant x$.

Observe that the definition of approximately parallel with separation and overlap ensures not only that line segments can vary slightly from being truly parallel and must be within a given orthogonal separation from each other but also that the orthogonal projection of one onto the other must intersect over a length corresponding to the overlap parameter. This rules out collinear line segments as if they overlapped to that extent they would have been vectorised into a single line segment. In the remainder we will refer to this concept only as approximately parallel.

3.3 Parameters

We now define a number of strict and fuzzy parameters that will be used in our rule system.

Radius of collinearity rc: Strict parameter used in the definition of *approximate collinearity*.

Dash length dl: Fuzzy parameter indicating the acceptable length of a single dash in a dashed line.

Dash separation ds: Fuzzy parameter indicating the acceptable distance between consecutive dashes in a dashed line. It is used for determining dash-neighbouring elements.

Observe that we require that rc is less than the minimum values of both dl and ds.

Bold dash length bdl: Fuzzy parameter indicating the acceptable length of a single line segment representing a bold dash in a dashed bold line (bold dashed bond).

Bold dash width bdw: Fuzzy parameter indicating the acceptable width of a single line segment representing a bold dash in a dashed bold bond.

Bold bond width bbw: Strict parameter indicating minimum acceptable width of a line segment representing a bold bond.

Wedge base wb: Fuzzy parameter indicating the acceptable width of a wedge base (for solid, hollow and dashed wedges).

Bond separation bs: Strict parameter indicating the acceptable maximum distance between two parallel bonds.

Minimal overlap ol: Strict parameter indicating the minimal overlap for two line segments to be considered overlapping.

Note, that bs and ol are the parameters that generally instantiate the definition of approximately parallel: $\|_{bs}^{ol}$.

4. RULES SYSTEM

The rule based system for the recognition of molecule structures (given in tabular form below) derives information on bonds only, not taking additional information on character groups into account. It rewrites a set of *geometric objects* to a

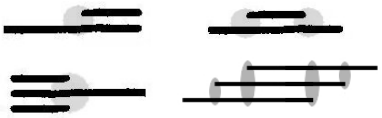

Figure 3. Implicit Nodes (circled)

Figure 4. Collinearity of Dashes in Dashed Bond

Figure 5. Dashes in Dashed Bold Bond

set of *abstract objects*, which are used for constructing a graph containing appropriate edges for each bond and nodes for each bond end, where bond ends connected to the same character groups share nodes.

A rule is defined in terms of preconditions and consequences. It is applicable if there exist geometric objects that satisfies its preconditions. Its consequence results in one or several new abstract objects as well as the removal of existing geometric objects and possibly the addition of new geometric objects. Preconditions of different rules are mutually exclusive, and thus the order of application is irrelevant. We are also not concerned here with how one concretely measures and implements the preconditions but will describe some details of a concrete implementation in the next section.

Rules R1, R2, and R3 deal with regular single, double, and triple planar bonds. However, this is less straight forward than it appears, as these bonds can not only occur separately, as displayed in Fig. 2(a)–2(c), but also can be given implicitly within single line segments. Fig. 3 demonstrates this issue, where we have a combination of single bonds with double and triple bonds, respectively. Carbon atoms are understood to be at the grey circled areas separating the bonds. As a consequence we need a mechanism to split up these implicit nodes to be used within the consequences of the three rules for planar bonds. We call this mechanism "*cutting*". Given a set of 2 or 3 approximately parallel line segments, we can define a minimal bounding rectangle that contains them. We define cut lines orthogonal to the long axis of the rectangle at the end points of the component line segments. Cut lines that are approximately the same (within rc of each other) can be unified. Now we can split each line segment at the cut lines into separate (collinear) line segments. With 2 line segments, there are at most two cut lines and precisely one section corresponding to a double bond. With 3 segments there can be up to four cut lines, but we can nevertheless precisely determine and cut the section corresponding to a triple bond.

The first three rules are sufficient to deal with all planar bonds, with the exception of those consisting of very short line segments. For example, the triple bond in Fig. 1 between N^+ and C^- could be incorrectly recognised as a bold dashed bond, as we are not taking character groups into account. A similar situation can arise for dashed wedge bonds. In order to avoid this, rules R2 and R3 have conditions to ensure a minimum line length, and thus the triple bond in Fig. 1 would not be captured. Instead we deal with the ambiguous cases in rules R4, R5 and R6. The issue here is that these cases correspond to one type of bond in one direction or a different type of bond in the orthogonal direction. To determine which type of bond it is, we need to identify which of the four sides of the bonds are connected to atoms in the molecule. Clearly, a side of the bond that meets the end of another bond or a character group, has such a connection. However, single carbon atoms are often omitted in molecule diagrams. In practice, whenever such ambiguous cases arise, there is always at least one side of the questionable bond with a bond connection or explicit atom or atom group, and often there are two at opposite sides of the bond. This is sufficient to identify the axis of the bond and therefore disambiguate it. However, this can only be done when all other bonds have been identified, after the rewrite process has terminated and the output graph is constructed.

Preconditions for rules R7, R8, and R9 are illustrated in Fig. 4–6, respectively. Hollow wedge bonds, depicted in Fig. 7 are dealt with in two separate cases — rules R10 and R11 — depending on whether or not their base is connected from the middle to another line segment. For wavy bonds the initial vectorisation returns a connected sequence of line segments arranged in a zig-zag pattern (Fig. 8), which is recognised by R12. Rules R13–R15 deal with single geometric primitives directly, whereas R16 recognises one particular drawing style of aromatic rings represented by large circles inside cyclic structures. See Fig. 9 for different versions of aromatic rings where the leftmost is the one captured by R16.

Finally bridge bonds are 3-dimensional structures where there are multiple different connection paths between different parts of the molecule. These are typically presented in a $2^1/_2$-dimensional perspective drawing form such as in Fig. 10. The bond lines that cross in the drawing but do not intersect in 3 dimensions are sometimes drawn in an *open* form (see Fig. 10(a)), where the background bond is drawn broken, and sometimes in a closed form (see Fig. 10(b)), leaving the reader to deduce that the crossing point does not correspond to a junction. Currently, these are handled as special cases in rules R17 and R18, respectively, since otherwise they are a set of single bonds which would be captured by rule R1.

Figure 6. Dashes in a Dashed Wedge Bond

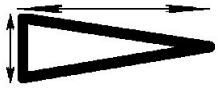

Figure 7. A Hollow Wedge Bond

Figure 8. Wavy Bond

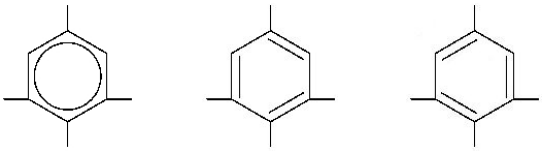

Figure 9. Aromatic Ring (these structures are equivalent)

(a) Open-bridge (b) Closed-bridge

Figure 10. Closed and open bridge bonds (circled)

R1: Single Planar Bond

1. l is a line segment.
2. There is no line segment l' such that $l \parallel_{bs}^{ol} l'$.

Cons.: l is a single bond with endpoints of l

R2: Double Planar Bond

1. $L = \{l_1, l_2\}$ is a set of two line segments.
2. $length(l_1) > wb$ and $length(l_2) > wb$.
3. $l_1 \parallel_{bs}^{ol} l_2$.
4. There is no $l \notin L$ such that $l_1 \parallel_{bs}^{ol} l$ or $l_2 \parallel_{bs}^{ol} l$.

Cons.: Cutting(l_1, l_2) will yield a double bond as well as at most two new line segments.

R3: Triple Planar Bond

1. $L = \{l_1, l_2, l_3\}$ is a set of three line segments.
2. for every $l \in L$ we have $length(l) > wb$.
3. $l_1 \parallel_{bs}^{ol} l_2$ and $l_2 \parallel_{bs}^{ol} l_3$ and $l_1 \parallel_{2*bs}^{ol} l_3$.
4. There is no $l \notin L$ such that $l_1 \parallel_{bs}^{ol} l$ or $l_2 \parallel_{bs}^{ol} l$ or $l_3 \parallel_{bs}^{ol} l$.

Cons.: Cutting(l_1, l_2, l_3) will yield a triple bond as well as at most four new line segments.

R4: Dashed Bold Bond vs Triple Bond

1. $L = \{l_1, l_2, l_3\}$ is a set of line segments.
2. Centre points of all $l \in L$ are approximately collinear.
3. $l_1 \parallel_{ds}^{bdl} l_2$ and $l_2 \parallel_{ds}^{bdl} l_3$.
4. There is no $l \notin L$ such that $l_1 \parallel_{ds}^{bdl} l$ or $l_2 \parallel_{ds}^{bdl} l$ or $l_3 \parallel_{ds}^{bdl} l$.
5. All elements of L have length of approximately bdl.
6. All elements of L have width of approximately bdw.

Cons.: Either dashed bold bond or triple bond.

R5: Dashed Wedge vs Triple Bond

1. $L = \{l_1, l_2, l_3\}$ is a set of line segments,
2. Centre points of all $l \in L$ are approximately collinear.
3. $l_1 \parallel_{ds}^{bdl} l_2$ and $l_2 \parallel_{ds}^{bdl} l_3$.
4. There is no $l \notin L$ such that $l_1 \parallel_{ds}^{bdl} l$ or $l_2 \parallel_{ds}^{bdl} l$ or $l_3 \parallel_{ds}^{bdl} l$.
5. $length(l_1) > length(l_2)$ and $length(l_2) > length(l_3)$.

Cons.: Either dashed wedge bond or triple bond group with implicit nodes.

R6: Dashed Wedge vs Double Bond

1. $L = \{l_1, l_2\}$ is a set of line segments.
2. Centre points of all $l \in L$ are approximately collinear.
3. $l_1 \parallel_{ds}^{bdl} l_2$.
4. There is no $l \notin L$ such that $l_1 \parallel_{ds}^{bdl} l$ or $l_2 \parallel_{ds}^{bdl} l$.
5. $length(l_1) > length(l_2)$.

Cons.: Either dashed wedge bond or double bond.

R7: Dashed Bond

1. $L = \{l_1, \ldots, l_n\}$, where $n \geq 3$, is a set of line segments,
2. L is approximately collinear.
3. Every element of L has length of approximately dl.
4. No two elements of L have a separation distance of less than the minimum of ds.
5. Two elements of L, called *end elements*, dash-neighbour wrt. ds precisely one other element of L. All other elements of L, called *internal elements*, dash-neighbour wrt. ds precisely two other elements of L.

Cons.: L forms dashed bond with endpoints given by endpoints of the minimal line segment that contains $l_1, \ldots, l_n$.

R8: Dashed Bold Bond

1. $L = \{l_1, \ldots, l_n\}$, where $n \geq 4$, is a set of line segments,
2. Centre points of all $l \in L$ are approximately collinear.
3. Every element of L has length of approximately bdl.
4. Every element of L has a width of approximately bdw.
5. No two elements of L have a separation distance of less than the minimum of ds.
6. Two elements of L, called *end elements*, dash-neighbour wrt. ds precisely one other element of L. All other elements of L, called *internal elements*, dash-neighbour wrt. ds precisely two other elements of L.

Cons.: L forms dashed bold bond with endpoints given by the endpoints of the minimal line segment that contains the centre points of the long sides of the line segments $l_1, \ldots, l_n$.

R9: Dashed Wedge Bond

1. $L = \{l_1, \ldots, l_n\}$, where $n \geq 4$, is a set of line segments.
2. Centre points of all $l \in L$ are approximately collinear.
3. No two elements of L have a separation distance of less than the minimum of ds.
4. Two elements $e_l, e_u \in L$ and $e_l < e_u \geq wb$, called the end elements, dash-neighbour precisely one other element of L. All other elements of L, called internal elements, neighbour precisely two other elements of L.
5. One neighbour of any internal element must be strictly longer than the internal element, the other neighbour must be strictly shorter.
6. Length of elements of L is strictly monotonically increasing from e_l to e_u.

Cons.: Dashed wedge bond from e_l to e_u with direction from e_l to e_u

R10: Hollow Wedge Bond - case 1

1. $L = \{l_1, l_2, l_3\}$, is a set of lines.
2. $l_1, l_2 \in L : length(l_1) = length(l_2)$.
3. $l_3 \in L : length(l_3) \in wb$.
4. $length(l_1) > length(l_3)$ and $length(l_2) > length(l_3)$.
5. For any $l_i \in L$ and $l_j \in L$, l_i and l_j have exactly one end point in common.

Cons.: Hollow wedge bond from joint end point of l_1, l_2 and centre point of l_3

R11: Hollow Wedge Bond - case 2

1. $L = \{l_1, l_2, l_3, l_4\}$, is a set of lines.
2. l_3 and l_4 are collinear and have one end point in common. Let l_5 be the minimal line segment that contains l_3 and l_4, and $L' = \{l_1, l_2, l_5\}$, is a set of lines.
3. $l_1, l_2 \in L' : length(l_1) = length(l_2)$.
4. $l_5 \in L' : length(l_3) \in wb$.
5. $length(l_1) > l_5$ and $length(l_2) > l_5$.
6. For any $l_i \in L'$ and $l_j \in L'$, l_i and l_j have exactly one end point in common.

Cons.: Hollow wedge bond with direction from joint end point of l_1, l_2 and centre point of l_5

R12: Wavy Bond

1. $L = \{l_1, \ldots, l_n\}$, where $n \geq 3$, is a set of line segments.
2. Centre points of all $l \in L$ are approximately collinear.
3. $\forall l \in L : length(l) \in dl$.
4. All elements of L are connected.
5. Two elements of L, called end elements, dash-neighbour precisely one other element of L. All other elements of L, called internal elements, neighbour precisely two other elements of L.
6. Two end points that are not connected must be the pair of end points that are furthest apart.

Cons.: A wavy bond between the furthest endpoints.

R13: Arrow Bond

1. l is an arrow.

Cons.: A dative bond from source to head.

R14: Solid Wedge Bond A solid wedge bond is handled as follows:

1. l is a solid triangle.

Cons.: A wedge bond with direction from tip to base.

R15: Bold Bond The following deals with a bold bond:

1. l is a line segment.
2. $width(l) > bbw$.

Cons.: A bold bond.

R16: Aromatic Ring

1. Let C be a circle.
2. $L = \{l_1, \ldots, l_n\}$, where $n \geq 5$, is a set of line segments.
3. All elements of L are within bs of C.

Cons.: n aromatic bonds, one for each $l_i \in L$.

R17: Open Bridge Bond

1. $L = \{l_1, l_2, l_3\}$ is a set of line segments.
2. l_1 and l_2 are approximately collinear.
3. $dist(l_1, l_2) \in ds$.
4. l_3 passes between the closest endpoints of l_1 and l_2.

Cons.: Replace l_1 and l_2 with the smallest line segment containing them.

R18: Closed Bridge Bond

1. $L = \{l_1, l_2, l_3, l_4\}$ is a set of line segments.
2. l_1 and l_2 are approximately collinear, as are l_3 and l_4.
3. $dist(l_1, l_2) = 0$ and $dist(l_3, l_4) = 0$.
4. l_1 and l_2 are part of an irregularly shaped cycle.
5. There is no character group where these lines connect.

Cons.: Replace l_1 and l_2 with the smallest line segment containing them, and similarly for l_3 and l_4.

5. IMPLEMENTATION

The input image is binarised and a connected component labelling is performed. The resulting connected components are fed into an OCR engine to identify character symbols which are in turn grouped to make character groups. This is done considering cases that require disambiguation such as capital "O", small case "l", capital case "I", etc. Any detected labels are erased from the image and the new image is thinned to unit width thickness. The lines, which can be free or

	OSRA Succeed		OSRA Failed		Total	
MolRec Succeed	4277	(74.58%)	796	(13.88%)	5037	(88.46%)
MolRec Failed	152	(2.65%)	510	(8.89%)	662	(11.54%)
Total	4429	(77.23%)	1306	(22.77%)	5735	(100%)

Table 1. Comparison of recognition results for OSRA dataset

	OSRA Succeed		OSRA Failed		Total	
MolRec Succeed	3823	(66.72%)	981	(17.12%)	4804	(83.84%)
MolRec Failed	335	(5.85%)	591	(10.31%)	926	(16.16%)
Total	4158	(72.57%)	1572	(27.43%)	5730	(100%)

Table 2. Recognition results for Maybridge dataset

connected forming polylines, are traversed and split at junctions where more than 2 lines intersect. An average line width is calculated during this stage. This results in a set of polylines that is used as input to the Douglas-Peucker line simplification algorithm.[17] The result of using this algorithm is a set of straight lines and an average line length.

Figure 11. Detection of Wedge/Bold Bonds

To extract precise geometric information about bonds in the shape of solid triangles, we use a disc of a radius larger than the previously measured average line thickness. This disc can fit inside the base of a solid triangle but not in a single bond line. We grow the disk until it reaches the largest size possible while still covering only black pixels in the original image. Then we walk the position of the disk in any direction that allows it to continue to grow. If this object is indeed a triangle, then when it can grow no more we have found the base of the triangle, thus identifying the stereo-centre atom position for this 3D bond. We can then walk the disk along in the direction of slowest decrease of disk size to find the opposite end of the bond. If it is not a triangle, then the disk size will not change appreciably over the length of the thinned line corresponding to this connected component. In this case we recognise the object as a thick solid line Fig. 11, identify the two end points and determine the stereo-centre heuristically as mentioned above.

Having obtained the geometric objects, the rule system is applied and an initial undirected graph is constructed where each bond is an edge and each junction is a node. This is done by grouping line segment endpoints by distance and by connectivity to the bounding box of atom or molecule names in order to construct each node. Each vertex of the graph is labelled with the atom or molecular sub-formula it represents. Bonds and atoms (edges and vertices) in the graph are examined for common causes of ambiguity. The graph is checked for open and closed bridge bonds. Also, disambiguation of lower case "l", capital case "I", the digit "1" and a vertical single bond is carried out at this stage. To generate output as MOL-files, character groups identifying more than one atom (called superatoms — the molecular diagram equivalent of macros or subroutines), have to be replaced with the molecule subgraph corresponding to that superatom. These superatom graphs are looked up in a dictionary of such structures. When the superatom has only one atom that is connected to the surrounding molecule, splicing the subgraph in is simple. However, if it has more than one such connection, then there are multiple ways it could be connected and these can only be resolved by expert chemical knowledge.

6. EXPERIMENTS

To evaluate our recogniser, we compared our system MolRec with OSRA,[5] the only system we are aware of that is freely available. We ran both systems on two different datasets: (1) OSRA's own benchmark set comprising 5735 computer generated diagrams with their corresponding MOL files and (2) a dataset we built from 5730 scanned molecule images taken from a catalogue of drug compounds by Maybridge.[18] The catalogue's pages were scanned at resolution of 300dpi and automatically segmented using a simple bespoke algorithm. This resulted in images of molecule structures with their corresponding Chemical Abstracts Service Registry numbers (extracted from the scanned images). These numbers were used to look up structures in online chemical databases to obtain their corresponding International Chemical Identifiers. These in turn were converted to MOL files using the open source tool, OpenBabel.[19]

Since both OSRA and MolRec can generate MOL file output we used OpenBabel to compare different MOL files semantically, ignoring unimportant syntactic differences. Since our initial experiments showed OSRA underperforming due to its limited superatom dictionary, we enhanced OSRA by adding all the superatom descriptions from the Marvin

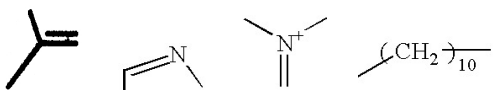

Figure 12. Various Touching Components

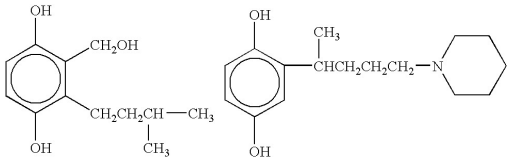

Figure 13. Unhandled Superatoms

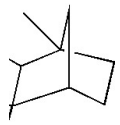

Figure 14. Ambiguous Open-bridge Bonds

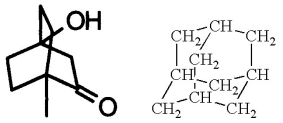

Figure 15. Ambiguous Connectivity of Atoms and Bonds

Abbreviation Group Collection[20] before doing the comparison. The results of our experiments for the two datasets are given in Tables 1 and 2, respectively. Observe that both tables denote the number of images on which (a) both systems succeed, by returning the correct MOL file as recognition result (top left corner), (b) both systems fail (bottom right), (c) either OSRA succeeds and MolRec fails (bottom left), or (d) MolRec succeeds and OSRA fails (top right).

The results demonstrate that MolRec generally outperforms OSRA, even on its own benchmark set. However, there is also a significant number of images where only one system succeeds. A subsequent analysis of recognition errors yielded the following most common problems:

Touching Components: The datasets have numerous images with touching components. These include touching characters (ligatures are very common), characters glued to symbols, characters touching bonds, symbols touching bonds and bonds touching bonds (Fig. 12). Both datasets were examined for touching components. The OSRA dataset has 1106 (19.29%) images with touching components (including ligatures), and 114 (1.99%) images excluding ligatures. On the other hand, the Maybridge dataset has 246 images with touching components (4.29%).

Superatoms: The second most common reason for recognition errors is in the system choosing an incorrect permutation of super atom connections for superatoms which have more than one connection to the main structure (Fig. 13).

Ambiguous Open-bridge Bonds: A number of open bridge bonds were very ambiguous, causing several images to be incorrectly recognised. The ambiguity was because one of the connecting parts of the bond was too short. (Fig. 14).

Other Failure Reasons: These include OCR classification errors, unhandled bond types, incorrect grouping of atoms, symbols and bonds. These also include a few very ambiguous images such as those shown in Fig. 15.

7. CONCLUSIONS

We have presented a rule based approach to molecule diagram recognition, that aims to provide a clear definition of the single cases that have to be distinguished during the recognition process. We have implemented our approach in a first prototype, which makes use of a conventional technique for line thinning followed by polyline simplification using the Douglas-Peucker algorithm before applying the rule based recognition to obtain a graph representation of the molecule. Nevertheless, this first prototype already achieves 88.46% and 83.84% recognition rates on two large datasets of diagrams. We also entered MolRec into the Image2Structure task of the TREC 2011 Chemistry track and achieved a recognition rate of 95% and 94.9% on two runs of 1000 diagrams.[16]

The experiments described identify obvious targets for improving our system, including correctly separating touching components, adding support for more bond types and improving our analysis of open bridge bonds. A significant source of errors is the connection permutation problem for superatoms, which will require deeper chemistry knowledge in our tool.

Early experiments gave clear evidence that the Hough transform approach favoured by some tools in this area was much harder to fine tune and not nearly as robust and accurate as our approach of line thinning followed by polyline simplification using the Douglas-Peucker algorithm. Our approach to identifying and orienting filled triangles has also proved to be a robust, efficient and accurate method. Finally, mining existing CTfiles has proved much more effective and useful than manually building SMILES[14] dictionaries for superatoms because of the extra information about superatoms that can be obtained in this manner.

REFERENCES

[1] McDaniel, J. and Balmuth, J., "Kekulé: Ocr-optical chemical (structure) recognition," *J Chem Inform Comput Sci* **32**(4), 373–378 (1992).

[2] McDaniel, J. and Balmuth, J., "Automatic interpretation of chemical structure diagrams," in [*Proc. of GREC-95*], 148–158, Springer (1996).

[3] Ibison, P., Jacquot, M., Kam, F., Neville, A. G., Simpson, R. W., Tonnelier, C. A. G., Venczel, T., and Johnson, A. P., "Chemical literature data extraction: The clide project," *J Chem Inform Comput Sci* **33**(3), 338–344 (1993).

[4] Valko, A. T. and Johnson, A. P., "CLiDE Pro: The latest generation of CLiDE, a tool for optical chemical structure recognition," *J. Chem. Inf. Model.* **49**(4), 780–789 (2009).

[5] Filippov, I. V. and Nicklaus, M. C., "Optical structure recognition software to recover chemical information: OSRA, an open source solution," *J. Chem. Inf. Model.* **49**(3), 740–743 (2009).

[6] Algorri, M. E., Zimmermann, M., and Hofmann-Apitius, M., "Automatic recognition of chemical images," in [*Proc. of ENC 2007*], 41–46 (2007).

[7] Algorri, M. E., Zimmermann, M., Friedrich, C. M., Akle, S., and Hofmann-Apitius, M., "Reconstruction of chemical molecules from images," in [*Proc. of EMBS 2007*], 4609–4612 (2007).

[8] Ramel, J., Boissier, G., and Emptoz, H., "Automatic reading of handwritten chemical formulas from a structural representation of the image," *Proc. of ICDAR'99* , 83, IEEE Computer Society, (1999).

[9] Ouyang, T. Y. and Davis, R., "Recognition of hand drawn chemical diagrams," in [*Proc. of 20th AAAI*], 846–851, AAAI Press (2007).

[10] Ouyang, T. and Davis, R., "Chemink: a natural real-time recognition system for chemical drawings," in [*Proc. of IUI '11*], 267–276, ACM (2011).

[11] Park, J., Rosania, G., Shedden, K., Nguyen, M., Lyu, N., and Saitou, K., "Automated extraction of chemical structure information from digital raster images.," *Chemistry Central* **3**(1) (2009).

[12] Schulenburg, J., "GOCR: Open source character recognition," (2008). `http://jocr.sourceforge.net/`.

[13] Nakagawa, K., Fujiyoshi, A., and Suzuki, M., "Ground-truthed dataset of chemical structure images in japanese published patent applications," in [*Proc. of 9th DAS*], 455–462, ACM (2010).

[14] Weininger, D., "SMILES, a chemical language and information system," *J Chem Inform Comput Sci* **1**, 31–36 (1988).

[15] Symyx, "CTfile formats," (2010). `http://www.symyx.com/downloads/public/ctfile`.

[16] Sadawi, N. M., Sexton, A. P., and Sorge, V., "Performance of MolRec at TREC 2011 — overview and analysis of results," in [*Proc. of 20th TREC*], NIST, (2011).

[17] Douglas, D. H. and Peucker, T. K., "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica* **10**(2), 112–122 (1973).

[18] "Maybridge Catalogue compounds for drug discovery," (2006). `http://www.maybridge.com/`.

[19] "Open Babel: The open source chemistry toolbox." `http://openbabel.org/`.

[20] "The Marvin abbreviation group collection." `http://www.chemaxon.com/marvin-archive/5.7.0.0/marvin/chemaxon/marvin/templates/default.abbrevgroup`.