TRUEVIZ: a groundtruth/metadata editing and visualizing toolkit for OCR

Tapas Kanungo[a], Chang H. Lee[a], Jeff Czorapinski[b] and Ivan Bella[b]

[a]Center for Automation Research, University of Maryland/College Park
[b]Lockheed Martin Management and Data Systems, 7100 Standard Drive, Hanover, MD 21076

ABSTRACT

Tools for visualizing and creating groundtruth and metadata are crucial for document image analysis research. In this paper, we describe TrueViz which is a tool for visualizing and editing groundtruth/metadata for OCR. TrueViz is implemented in the Java programming language and works on various platforms including Windows and Unix. TrueViz reads and stores groundtruth/metadata in XML format, and reads a corresponding image stored in TIFF image file format. Multilingual text editing, display, and search module based on the Unicode representation for text is also provided. This software is being made available free of charge to researchers.

Keywords: annotation, groundtruth, visualization, multilingual, multiplatform, Java, XML, OCR

1. INTRODUCTION

Groundtruth data is crucial for document image analysis because it is quite impossible to train and test algorithms without it. Since groundtruth is created manually in most cases, tools for annotating and visualizing groundtruth are very important. At the MLOCR99 workshop[1] the consensus in the corpus working group was that our community needs i) a protocol for groundtruthing documents, ii) an XML-based groundtruth representation format, iii) a public domain multilingual/multiplatform visualization and dataentry tool, iv) a consortium for managing and distributing the datasets. In this paper we address two of the four issues raised: i) we describe an XML-based groundtruth represenation format, and ii) we describe, TrueViz, which is a public domain annotation tool that we developed at the University of Maryland, College Park.

This paper is organized as follows. In Section 2 we describe various existing annotation tools used in the document image analysis and related areas like speech recognition, linguistics, and information retrieval. The main features of our TrueViz toolkit are described in Section 3. In Section 4 we discuss the design and implementation issues related to editing, visualization, and search. The XML data format for groundtruth is discussed in Section 5. Here we also provide representative samples of XML files. The multilingual data entry, visualization, and search features of TrueViz are quite unique and are discussed in Section 6. Finally, in Section 7 we list the things that we plan to incorporate into TrueViz in future.

2. PREVIOUS WORK

There are many annotation and visualization tools in various domains. In this section we describe a few annotation tools commonly used in document image analysis, speech recognition, linguistics, information retrieval, video analysis, geographic system, and statistics. In Table 1, we provide a comparison of these tools.

Further author information: (Send correspondence to T.K.)
T.K.: E-mail: kanungo@cfar.umd.edu
C.H.L.: E-mail: chlee@cs.umd.edu

Table 1. Comparison of Visualization Tools

Name	Platform	Data Format	Domain
PinkPanther	Unix/X Windows System	ASCII	Document Image Groundtruth
Illuminator	Unix/X Windows System	DAFS	Document Image Groundtruth
Oulu Database Browser	Multi-Platform/Java	ASCII	Document Image Groundtruth
TrueViz	Multi-Platform/Java	XML Format	Document Image Groundtruth
Transcriber	Unix/Windows NT	XML Format	Speech Annotation
ATLAS	Unix/Windows NT	XML Format	Linguistic Annotation
Alembic Workbench	Unix system	SGML/PTF Format	Linguistic/Named Entities Annotation
ViPER	Multi-Platform/Java	ASCII	Video Sequence Groundtruth
XGobi	Unix/X Windows System	S Data Format/ASCII	Statistical Data
S-PLUS	Windows 95/98	Customized Data	Statistical Data
CLASP	Unix/Macintosh	Commonly Used Formats	Statistical Data
Mondrian	Multi-Platform/Java	ASCII/Databases	Categorical /Geographical Data
PolyPaint+	SunOS/Solaris	netCDF	Geographic Data
Spotfire	MS Windows	Database/Spreadsheet /ASCII	Decision Making by Data Analysis
Slicer Dicer	MS Windows	Binary/ASCII /Commonly Used Formats	Medical/Scientific Data Defined on Grids

2.1. Document Visualization Tools

Visualization tools for displaying or editing a document image and groundtruth metadata have been developed for evaluating various algorithms, creating document groundtruth, or browsing documents.

Pink Panther[2] is an environment for creating segmentation groundtruth files and page segmentation benchmarking. Page segmentation is the process of decomposing a document page image into its structural and logical units, such as images, paragraphs, headlines, tables, etc. The performance of a page segmentation algorithm is evaluated by running the algorithm on a set of document images, and comparing the output for each document to corresponding goundtruth metadata. Pink Panther consists of two parts: Grounds-Keeper and Cluzo. Grounds-Keeper is a tool for creating groundtruth metadata. Grounds-Keeper visualizes a document image and document metadata. It also allows users to zone the document image and specify the information for each zone. Groundtruth metadata created by Grounds-Keeper is stored in an ASCII file format. Cluzo is a benchmarking tool for collecting the location, type and severity of the segmentation errors on the page as well as the information on the segmentation performance. Pink Panther is implemented on the Unix/X Windows platform and written in C. While Grounds-Keeper allows the user to enter segmentation groundtruth, entering text groundtruth is not possible.

Illuminator[3] is an editor developed by RAF Technology, Inc. for building document understanding test and training sets, for correction of OCR (Optical Character Recognition) errors, and for reverse-encoding the essential information and attributes of a document. Illuminator visualizes or edits a document image and its entities, which are specific regions of image and associated metadata. It is configured to handle text in major European languages and Japanese. Illuminator uses a DAFS (Document Attribute Format Specification)[3] file format to store document image and metadata. DAFS provides a format for breaking down a document into entities which have hierarchical structure, and for defining entity boundaries and attributes. Illuminator is implemented on the Unix/X Windows platform and written in C.

The MediaTeam Oulu Document Database[4] is a collection of scanned documents with corresponding groundtruth for the physical and logical structure of the documents. It was developed by the University of Oulu MediaTeam. Document database browser is a visualization tool for exploring the contents of the document database. The browser is written in the Java programming language and one can visualize document images and corresponding metadata simultaneously. The browser can explore the database and select a particular document for visualizing the document. The browser also provides a window to list attributes of the document. Document images, which were originally stored in TIFF image format, are stored in JPEG image format and metadata is stored in an ASCII file format.

Pink Panther and Illuminator work only on the Unix platform. Because there are many tools that are executable only on the Windows system, Pink Panther and Illuminator have a limitation. Oulu document database browser is developed in Java programming language, and can be run in various platforms. However, Oulu document database supports JPEG image format only while TIFF image format is the most popular image format for document images. Furthermore, the file representation of the the groundtruth is non-standard. In fact, all the above tools store document metadata in their own file format. To provide data compatibility, a standard file format or the file format to which other file formats can be easily converted would be needed.

2.2. Other Visualization Tools

We surveyed existing visualization tools on other data domain to find out the best way to provides multi-platform and data compatibility. We will summarize features of visualization tools on various domains such as statistical, categorical, geographical, and medical data as well as linguistic data or speech signals.

Transcriber[5–7] is a tool for segmenting, labeling and transcribing speech signals. Transcriber allows support for most common audio formats and stores the transcription in XML format. It is developed with Tcl/Tk and C in Unix system and Windows NT system.

ATLAS[8] is an architecture and tool for linguistic analysis based on a formal model for annotating linguistic artifacts. ATLAS uses XML-based ATLAS Interchange Format(AIF) for storing annotated corpora, and it is developed with various programming languages such as C++, Perl, Tcl/Tk and Java.

Alembic Workbench[9] is a new set of integrated tools that uses a mixed-initiative approach to bootstrapping the manual tagging process with the goal of reducing the overhead associated with corpus development. Alembic Workbench is developed with Tcl/Tk, Perl, C and Lisp in Unix system, and it uses SGML and PTF (Parallel Tag File) format for source text and annotations.

ViPER (Video Processing Evaluation Resource)[10] consists of three main components: ViPER-GT, ViPER-PE, and ViPER-Viz. ViPER-GT contains modules for configuring and producing groundtruth information which describes a video sequence. ViPER-PE module provides performance evaluation capabilities for comparing computed results with appropriate groundtruth information. ViPER-Viz enables a user to visualize groundtruth, analysis results, performance evaluation results, or an entire video clip. ViPER is developed in Java programming language, and groundtruth and results are stored in ASCII file format.

XGobi[11–13] is an X Window application for interactively exploring statistical data. Current functionality includes brushing, identification, and the editing of connected lines, as well as rotation and the grand tour, with several interactive projection pursuit indices. Several functions can be linked so that actions in one window are promptly reflected in another.

S-PLUS[14] is the premier desktop data analysis tool for business and technical professionals who need sophisticated data analysis and visualization capabilities to identify trends and discern hidden meaning in data, enabling knowledge-driven decision making. It allows data import and export from leading spreadsheets such as Excel, as well as from a wide range of relational and other data sources.

The Common Lisp Analytical Statistics Package (CLASP)[15] is a tool for visualizing and statistically analyzing data. CLASP provides an interactive environment for data manipulation and statistical analysis and all the descriptive and hypothesis-testing statistics one expects of a moderately powerful statistics package. Furthermore, it includes many features that facilitate exploratory data analysis.

Mondrian[16] is a data-visualization system written in JAVA. The main emphasis of Mondrian is on visualization techniques for Categorical Data and Geographical Data. Mondrian provides various plots like Mosaic Plot, Maps, Barcharts and Parallel Coordinates, which are fully linked and offer various interrogations.

PolyPaint+[17] is an interactive scientific visualization tool that displays complex structures within three-dimensional data fields. It provides both color shaded-surface display and simple volumetric rendering in either index or true color. PolyPaint+ routines first compute the polygon set that describes a desired surface within the 3D data volume, and then these polygons are rendered as continuously shaded surfaces. Objects rendered volumetrically may be viewed along with shaded surfaces. Additional data sets can be overlaid on shaded surfaces by color coding the data according to a specified color ramp.

Spotfire[18] is the decision analytics workspace that uses the connectivity of the Web to provide scientific and industrial decision-making communities a unified workspace to access large amounts of complex data from wherever it resides and to visually explore and analyze data and to share results easily across an extended enterprise.

Slicer Dicer[19] is designed for scientists, medical professionals, engineers, and other technologists who need the analysis, interpretation and documentation of complex data defined in three or more dimensions. Slicer Dicer helps exploring the data visually by "slicing and dicing" to create arbitrary orthogonal and oblique slices, rectilinear blocks and cutouts, isosurfaces, and projected volumes. Slicer Dicer also provides animation sequences featuring continuous rotation, moving slices, blocks, parametric variation (time animation), oblique slice rotation, and varying transparency. More detailed review and taxonomy for visualization tools can be found in an article by Shneiderman.[20]

3. FEATURES

TrueViz currently has the following features:

Multiplatform: The tool is being developed using the Java programming language and works on the UNIX and Windows platforms. We have currently tested it on SUN Solaris 2.6 and Windows 2000. Other groundtruth generation tools like Illuminator and PinkPanther are UNIX based. Since our tool works on both platforms, windows users will spend less time in learning how to use the tool.

XML-based Representation: The page layout groundtruth is represented using the XML markup language, which is the current industry standard. The representation uses an attribute-value scheme and users can add attributes easily.

Entities: Currently we can visualize and edit page, zone, line, word and character level groundtruth. For each entity the user can create the attributes (e.g. bounding boxes) and specify the values.

Multilingual Visualization: Users can visualize groundtruth text in various languages since we allow users to represent text in Unicode.

Multilingual Data Entry: While regular English text can be entered by the regular keyboards, keyboard mapping for allowing other languages are also available. Currently we provide Russian (Cyrillic) keyboard and text input facility.

Converters: We proved converts to convert the University of Washington dataset in DAFS format to our XML representation. Other converters will have to be written to convert data existing in other formats such as XDOC into our format.

Logical information: The visualization tool allows user to visualize the logical reading order.

Search: Users can search for strings in the groundtruth and find the locations where it appears on the image. The search module works in any language since it uses the Unicode text representation. Furthermore, we allow users to specify edit distances for approximate searching.

Color: Entities with different attributes can be displayed in different colors. For example, images zones can be shown in one color and table or text zones in another.

Scale: The image and the overlaid groundtruth can be scaled up and down to various resolutions.

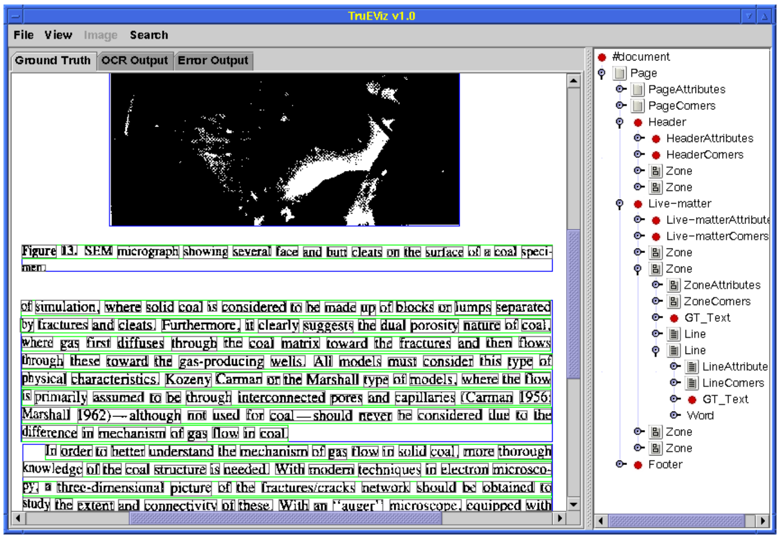


Figure 1. TrueViz is consists of an image panel (left panel) and a tree view(right panel).

4. DESIGN AND IMPLEMENTATION

4.1. Overview

TrueViz has two vertically splitted panels (see Figure 1). The left panel is an image panel for displaying a document image and corresponding geometric metadata, and the right panel is a tree view for displaying textual metadata structure.

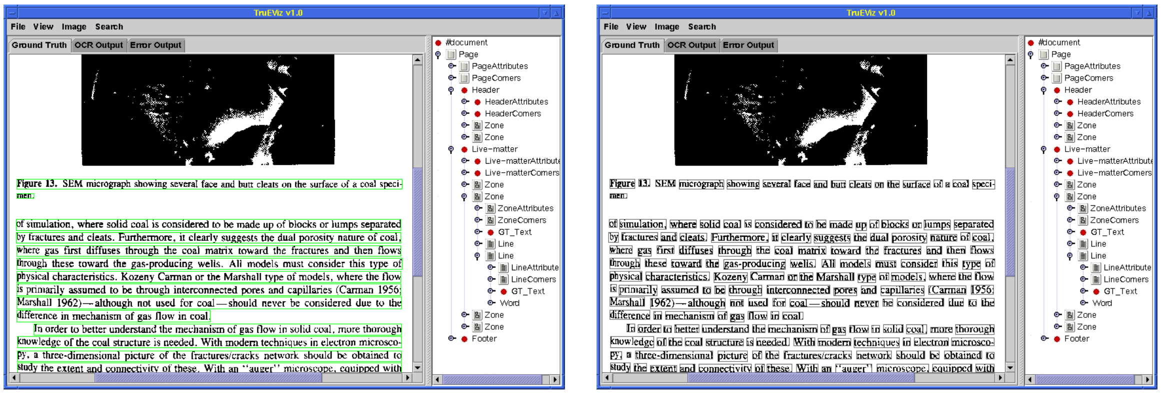
The image panel displays a document image and overlays geometric metadata on the image. Currently, three kinds of geometric metadata can be visualized: Bounding boxes, the logical relationship and Infopanel. The bounding boxes of an entity is visualized as a polygon whose color represents the type of the entity. The logical relationship means the logical reading order of the document entities, and it is visualized using an arrow from one entity to the next. The Infopanel is a small window for displaying few important attributes of the entity. Moreover, the image and metadata visualization can be scaled to various resolutions.

The tree view displays the XML-based groundtruth metadata. This viewer displays the data in a tree structure of expandable and collapsible nodes. The attribute values can be edited in the tree node and groundtruth text can be edited in the separate multilingual text editor.[21,22]

4.1.1. Metadata Visualization

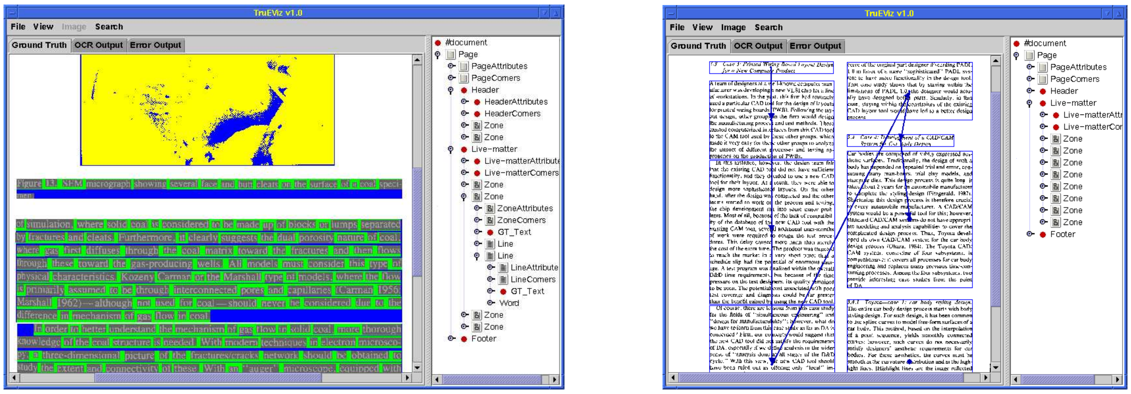
Entities can be classified into four categories: Zones, Lines, Words and Characters. Entities are hierarchal in nature so a Zone is contained within a Page, a Line is contained within a Zone, a Word is contained within a Line and a Character is contained within a Word. Because of the hierarchical nature of the entities, it is necessary to change views in order to view specific portions of that structure. There are five views: Image Only, Page, Zone, Line, Word and Character. The Image Only view shows only the image without any groundtruth visualization. The Page view shows metadata for all entities from the highest level to the lowest level. This view is not editable or selectable. The Zone view shows only Zone metadata. You may access a Zone's data by clicking on the Zone. This will cause the Zone to be active (selected) and highlighted, and the Infopanel to be poped up. The Infopanel is a small window for displaying important metadata for the active entity (see Figure 7). The corresponding node in the tree viewer will also be selected. Similarly, the Line view shows all Line metadata (see Figure 2 (a)), and the Word view shows all Word metadata (see Figure 2 (b)), and the Character view shows all Character metadata. Like the Zone view, metadata can be selected and the Infopanel for the active entity is poped up in the Line, Word and Character views.

There are two options for views: 'Fill Bounding Boxes' and 'Logical Relations'. If you check the 'Fill the Bounding Boxes' option, all entities are painted in colors corresponding to their types (see Figure 3 (a)). Otherwise, entities



(a) Line view displays all Line entities. (b) Word view displays all Word entities.

Figure 2. Hierarchical Display.



(a) Fill Bounding Boxes. (b) Logical Relations.

Figure 3. View options

are displayed using polygonal outlines whose colors also represents their types. This option will be useful when the document is displayed in a large scale because a user can see the type of a certain entity from its color even if the bounding box is too large to fit in a screen. If you select 'Show Logical Relation' option, the logical ordering relations are visualized (see Figure 3 (b)). The logical ordering relations are visualized using arrows from an entity to the next logical entity. This logical order is the reading order.

4.1.2. Metadata Editing

Groundtruth metadata can be edited in two ways: graphical editing and text editing. All metadata can be edited within the attribute value node in the tree view. Because the groundtruth text may contain multilingual text, the groundtruth text is edited in the separate multilingual text editor. The metadata visualized in the image panel can be edited graphically. It is very difficult to correct bounding boxes of entities by editing the coordinates. Therefore, TrueViz enables users to change the coordinates of bounding boxes graphically. As well as the bounding boxes, the logical relationship can be changed graphically. In addition, the image panel is used to create and delete entities.

4.2. Search

TrueViz provides a multilingual approximate search functionality. A search string and edit distance can be specified in the search window. TrueViz provides multilingual input methods for a search string. The edit distance is the minimum number of substitutions, insertions and deletions required to transform one string into the other. The

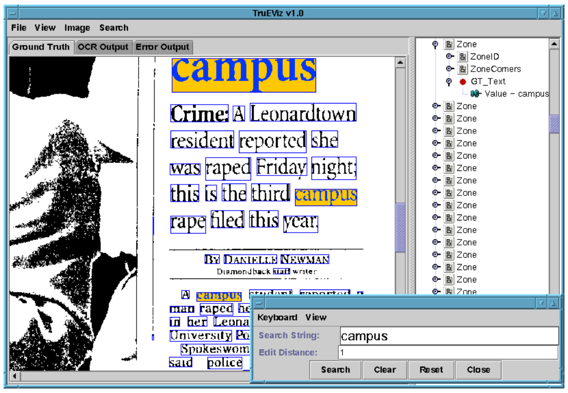


Figure 4. Search string "campus" and edit distance 1 are specified in the search window (in right lower corner), and the matching entities are highlighted.

maximum edit distance allowed during the search can be specified.[23] After the search is finished, all entities containing the search string with the specified edit distance are highlighted (see Figure 4).

5. THE DATA FORMAT

5.1. Overview

The groundtruth metadata is stored in the XML file format[24–27] (see Figure 5), and the document images are stored in the TIFF image file format. The tree view reflects the XML data file, and an internal data structure is created to visualize the groundtruth metadata. The internal data structure consists of Region of Interest (ROI) nodes. A ROI is a generic term used to describe any area of the image that the user deems of interest. The internal data forms a directed acyclic graph with ROIs as its nodes and hierarchical or logical links as its edges.

5.2. XML Data Format

The groundtruth data is organized in a hierarchical structure. The highest level and therefore most inclusive entity is the Document. A Document is, in its simplest form, a collection of individual units, known as Pages, which are related to or support a specific topic or purpose (e.g. a report or manual). A Page is a level down in the hierarchy and represents the individual units of a Document. Each Page has an associated image that represents the original hard copy. A Page contains one or more Zones. A Zone is usually a rectangular area definable by its horizontal and vertical coordinates within a page. The purpose of the Zone is to identify key areas of the page such as titles, headings, graphics, page numbers, etc. Each Zone may contain one or more Lines. A Line is an individual line of text. A Line could be broken down into one or more Words, each of which may contains one or more Characters. Each tag in the XML file represents an entity or its attribute. Even though the entity name can be any alphanumerical word, the entities which can be graphically edited in the TrueViz are Zone, Line, Word and Character.

The entity attributes can be listed under the entity tag in the XML file. While any attribute name can be listed, there are some built-in attributes which are crucial for the visualization of the groundtruth data.

ID: ID is the identification of the entity. The attribute name for ID is combined with the entity name. For example, ID of a Zone entity is represented as ZoneID, and LineID for Line, WordID for Word and CharacterID for Character.

Corners: Corners represents the bounding box of the entity. Left upper, right upper, right lower, left lower vertices are listed inside a Corners tag in order. Like ID, the attribute name is combined with the entity name.

Next: Next stores the ID of the logically following entity. Like ID, the attribute name is combined with the entity name.

GT_Text: GT_Text stores the groundtruth text of the entity.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Page SYSTEM "Trueviz.dtd">
<Page>
	<PageID Value="P000"> </PageID>
	<PageType Value="Journal"> </PageType>
	<PageNumber Value="1"> </PageNumber>
	<PageColumns Value="1"> </PageColumns>
	<Font Size="9-12" Spacing="Undefined" Style="Normal" Type="Serif"> </Font>
	<Zone>
		<ZoneID Value="Z000"/>
		<ZoneNext Value="Z001"/>
		<CharacterOrientation Type="String" Value="up-right"/>
		<DominantFontSize Type="String" Value="9-12"/>
		<DominantFontSpacing Type="String" Value="proportional"/>
		<DominantFontStyle Type="String" Value="plain"/>
		<DominantFontType Type="String" Value="serif"/>
		<Language Type="String" Value="English"/>
		<TextAlignment Type="String" Value="justified"/>
		<TextReadingDirection Type="String" Value="left-right"/>
		<ZoneCorners>
			<Vertex x="1281" y="3136"></Vertex>
			<Vertex x="1296" y="3136"></Vertex>
			<Vertex x="1296" y="3169"></Vertex>
			<Vertex x="1281" y="3169"></Vertex>
		</ZoneCorners>
		<GT˙Text Value="a"></GT˙Text>
		<Line>
			<LineID Value="Z000L000"/>
			<LineCorners>
				<Vertex x="1281" y="3136"></Vertex>
				<Vertex x="1296" y="3136"></Vertex>
				<Vertex x="1296" y="3169"></Vertex>
				<Vertex x="1281" y="3169"></Vertex>
			</LineCorners>
			<GT˙Text Value="a"></GT˙Text>
			<Word>
				<WordID Value="Z000L000W000"/>
				<WordCorners>
					<Vertex x="1281" y="3136"></Vertex>
					<Vertex x="1296" y="3136"></Vertex>
					<Vertex x="1296" y="3169"></Vertex>
					<Vertex x="1281" y="3169"></Vertex>
				</WordCorners>
				<GT˙Text Value="a"></GT˙Text>
				<Character>
					<CharacterID Value="Z000L000W000C000"/>
					<CharacterCorners>
						<Vertex x="1281" y="3136"></Vertex>
						<Vertex x="1296" y="3136"></Vertex>
						<Vertex x="1296" y="3169"></Vertex>
						<Vertex x="1281" y="3169"></Vertex>
					</CharacterCorners>
					<GT˙Text Value="a"></GT˙Text>
				</Character>
			</Word>
		</Line>
	</Zone>
	<Zone>
		<ZoneID Value="Z001"/>
		<ZoneNext Value=""/>
		<CharacterOrientation Type="String" Value="up-right"/>
		<DominantFontSize Type="String" Value="9-12"/>
		<DominantFontSpacing Type="String" Value="proportional"/>
		<DominantFontStyle Type="String" Value="italic"/>
		<DominantFontType Type="String" Value="serif"/>
		<Language Type="String" Value="English"/>
		<TextAlignment Type="String" Value="justified"/>
		<TextReadingDirection Type="String" Value="left-right"/>
		<ZoneCorners>
			<Vertex x="2281" y="3136"></Vertex>
			<Vertex x="2296" y="3136"></Vertex>
			<Vertex x="2296" y="3169"></Vertex>
			<Vertex x="2281" y="3169"></Vertex>
		</ZoneCorners>
		<GT˙Text Value="b"></GT˙Text>
	</Zone>
</Page>
```

Figure 5. An XML example file.

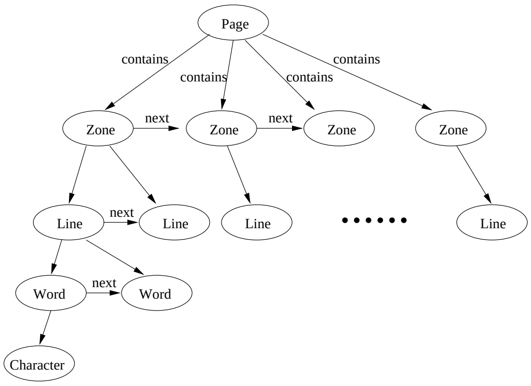


Figure 6. Entity Structure.

The following example shows a simple entity.

```
<Zone>
  <ZoneID Value="Z001"/>
  <ZoneNext Value="Z002"/>
  <GT_Text Value="Hello, world">
  <ZoneCorners>
    <Vertex x="10" y="10"/>
    <Vertex x="100" y="10"/>
    <Vertex x="100" y="30"/>
    <Vertex x="10" y="30"/>
  </ZoneCorners>
<Zone>
```

5.3. Internal Data Structure

The groundtruth metadata is stored in XML file format which is essentially a tree. However, entities form a directed acyclic graph structure. Each entities contains child entities and has a next logical entity. The graph representing the entity structure can be expressed by the equation (1) (see Figure 6).

$$G = (V, E) \quad where \quad V = \{Zone, Word, Line, Character\}, \quad E = \{contains, next\} \tag{1}$$

Because of the difference between the entity structure and the XML structure, TrueViz has an internal data structure that is a little different from the XML structure. The internal data structure consists of Region of Interest (ROI) nodes, and the ROIs forms a directed acyclic graph described in the equation (1). A next logical entity is stored as an attribute of an entity in the XML file, and is converted into a link from a ROI to the next ROI in the internal data structure.

6. MULTILINGUAL FEATURES

6.1. Multilingual Text Data

Java programs running on JDK1.1 or JDK1.2 can display any Unicode[28] character which can be rendered with a host font, and TrueViz displays multilingual text using Java Unicode facilities (see Figure 7). TrueViz can read Unicode characters from the XML file, and saves the XML file in the Unicode UTF8 format.[28] However Java does not provide a multilingual input method. We developed a multilingual input method, which is described in Section 6.2.

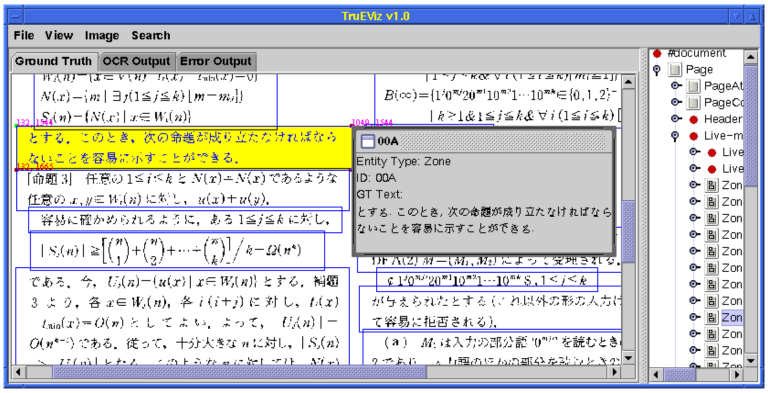
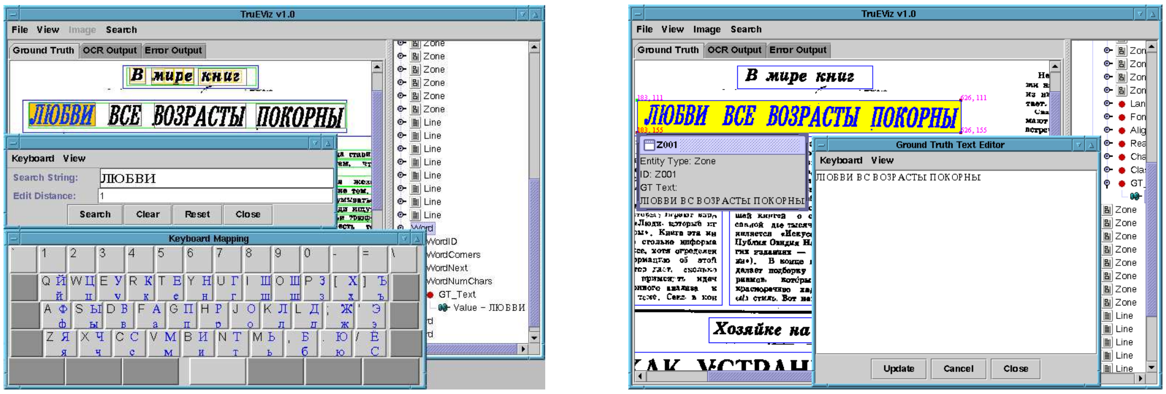


Figure 7. Infopanel and Multilingual Display.



(a) Russian Input Method in Search. For the users who are not familiar with the keyboard mapping, a keyboard mapping display window is provided.

(b) Russian Input Method in Groundtruth Editor.

Figure 8. Multilingual Input Method.

6.2. Multilingual Input Method

TrueViz provides a multilingual input system. The number of characters used in some languages like Chinese, Japanese, or Korean is bigger than the number of characters that can be input by a regular keyboard. In such languages, a sequence of several characters needs to be typed for constructing a character. While this composition process is going on, the input method accepts a sequence of characters, and produces composed text and committed text. The composed text is the intermediate text which is being processed to produce the intended text. The final input text is called committed text (see Figure 9). Input methods for various languages can be easily added using this common interface where the system accepts an input sequence and returns the composing text and the committed text. Besides the default English language input method, a Russian input method is currently implemented. Other input methods can be added by adding input method classes. The multilingual input methods can be applied anywhere multilingual text input is needed (see Figure 8). For example, TrueViz supports multilingual text input in the search window for multilingual search. For the people who you are not familiar with the keyboard mapping, TrueViz provides the keyboard mapping display which shows the keyboard mapping. In addition to the keyboard input system, TrueViz provides a Unicode character input using a code table so that any Unicode character can be selected and inserted into a text.

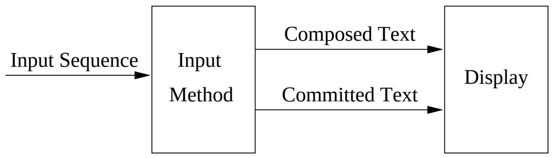


Figure 9. Input Method.

7. FUTURE WORK

TrueViz was tested by several colleagues within our research group, and we intend to do a more thorough quantitative user evaluation using questionnaires. The main issue raised by users was the speed, and so speed enhancement is one of the topics we will work on. The current entity structure is fixed to zone, line, word and character, and only these entities can be added or edited graphically. However, users may want to build a document metadata in their own structure. Therefore a flexible entity structure where a user defined entity structure is allowed will be very useful. Currently TrueViz provides only English and Russian multilingual input methods. Korean, Japanese, Chinese and other languages will also be made available in the future. Currently we have DAFS to XML and XML to DAFS converters. However we intend to provide converters from SGT format (produced by Pink Panther groundtruthing tool), XDOC (Xerox representation for groundtruth), and Caere representation. Document images have a huge data size, and XML files would require more disk space than binary-formatted files. Therefore, if compressed XML files can be saved and read, then the file size of the XML files would not be a concern. We plan to make the XML representation compatible with the performance evaluation toolkit PSET,[29–32] so that it is easy to visualize segmentation evaluation results using TrueViz.

ACKNOWLEDGMENTS

Authors would like to thank the participants of MLOCR99 and Glen van Doren of the Department of Defense for valuable input. This research was funded in part by the Department of Defense under Contract MDA0949-6C-1250, Lockheed Martin under Contract 9802167270, the Defense Advanced Research Projects Agency under Contract N660010028910, and the National Science Foundation under Grant IIS9987944. We would also like to thank Song Mao of University of Maryland for discussion and testing.

REFERENCES

1. "Working group notes," in *International Workshop on Performance Evaluation Issues in MultiLingual OCR, Bangalore, India*, September 1999. `http://www.cfar.umd.edu/~kanungo/workshop/reco.html`.
2. B. A. Yanikoglu and L. Vincent, "Pink panther: A complete environment for ground-truthing and benchmarking document page segmentation," *Pattern Recognition* **31**, pp. 1191–1204, 1998.
3. T. Fruchterman, "Dafs: A standard for document and image understanding," in *Proceedings of Symposium on Document Image Understanding Technology, Bowie*, pp. 94–100, October 1995.
4. J. Sauvola and H. Kauniskangas, *MediaTeam Oulu Document Database*. MediaTeam, University of Oulu, 1998. `http://www.mediateam.oulu.fi/MTDB/`.
5. C. Barras, E. Geoffrois, Z. Wu, and M. Liberman, "Transcriber: development and use of a tool for assisting speech corpora production," *Speech Communication special issue on Speech Annotation and Corpus Tools* **33**, January 2000.
6. E. Geoffrois, C. Barras, S. Bird, and Z. Wu, "Transcribing with annotation graphs," in *Second International Conference on Language Resources and Evaluation, Athens, Greece*, pp. 1517–1521, May-June 2000.
7. C. Barras, E. Geoffrois, Z. Wu, and M. Liberman, "Transcriber: a free tool for segmenting, labeling and transcribing speech," in *First International Conference on Language Resources and Evaluation, Granada, Spain*, pp. 1373–1376, May 1998.
8. S. Bird, D. Day, J. G. J. Henderson, C. Laptun, and M. Liberman, "ATLAS:a flexible and extensible architecture for linguistic annotation," in *Proceedings of the Second International Language Resources and Evaluation Conference, Athens, Greece*, pp. 1699–1706, May 2000.

9. D. Day, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson, and M. Vilain, "Mixed-initiative development of language processing systems," in *Fifth Conference on Applied Natural Language Processing, Washington, D.C*, March-April 1997.
10. D. Doermann and D. Mihalcik, "Tools and techniques for video performance evaluation," in *15th International Conference on Pattern Recognition, Barcelona, Spain*, pp. 167–170, September 2000. `http://documents.cfar.umd.edu/LAMP/Media/Projects/ViPER/`.
11. D. F. Swayne, D. Cook, and A. Buja, "Xgobi: Interactive dynamic data visualization in the X window system," *Journal of Computational and Graphical Statistics* **7**, 1998. `http://www.research.att.com/areas/stat/xgobi/`.
12. D. F. Swayne, N. Hubbell, and A. Buja, "XGobi meets S: integrating software for data analysis," in *Proceedings of Symposium on Interface*, pp. 430–434, 1991. `http://www.research.att.com/areas/stat/xgobi/`.
13. D. F. Swayne, D. Cook, and A. Buja, "XGobi: interactive dynamic graphics in the X window system with a link to S," in *Proceedings of the American Statistical Association Meetings*, 1992. `http://www.research.att.com/areas/stat/xgobi/`.
14. W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S-Plus*, Springer, 1999. `http://www.splus.mathsoft.com/`.
15. S. D. Anderson, D. L. Westbrook, A. Carlson, D. M. Hart, and P. R. Cohen, *Common Lisp Analytical Statistics Package: User Manual*, University of Massachusetts, 1995. `http://eksl-www.cs.umass.edu/clasp.html`.
16. University of Augsburg, *Mondrian*. `http://jetta.math.uni-augsburg.de/Mondrian/`.
17. National Center for Atmospheric Research/Mesoscale and Microscale Meteorology, *PolyPaint User Manual - Version 3.0*. `http://lasp.colorado.edu/polypaint/home.html`.
18. C. Ahlberg and B. Shneiderman, "Visual information seeking: Tight coupling of dynamic query filters with starfield displays," in *Proceedings of ACM CHI94 Conference, Boston*, pp. 313–317, April 1994. `http://www.spotfire.com/products/spotfire_net.asp`.
19. PIXOTEC, LLC., *Slicer Dicer*. `http://www.slicerdicer.com/`.
20. B. Shneiderman, "The eyes have it: A task by data type taxonomy of information visualizations," in *Proceedings of IEEE Symposium on Visual Languages*, pp. 336–343, September 1996. `http://otal.umd.edu/Olive/`.
21. B. Shneiderman, *Designing the User Interface*, Addison Wesley, 1998.
22. C. H. Lee and T. Kanungo, *TRUEVIZ User's Manual*, Language and Media Processing Laboratory, Center for Automation Center, University of Maryland, August 2000.
23. D. Gusfield, *Algorithms on strings, trees, and sequences: computer science and computational biology*, Cambridge University Press, 1997.
24. E. R. Harold, *XML Bible*, IDG Books, 1999.
25. T. Bray, J. Paoli, and C. M. Sperberg-McQueen, *Extensible Markup Language (XML)*. W3C, 1998. `http://www.w3.org/TR/REC-xml`.
26. L. Wood, A. L. Hors, V. Apparao, L. Cable, M. Champion, J. Kesselman, P. L. Hegaret, T. Pixley, J. Robie, P. Sharpe, and C. Wilson, *Document Object Model (DOM)*. W3C, 1999. `http://www.w3.org/TR/DOM-Level-2/`.
27. B. McLaughlin, *Java and XML*, O'Reilly, 2000.
28. T. U. Consortium, *The Unicode Standard, Version 2.0*, Addison Wesley Developers Press, 1997.
29. S. Mao and T. Kanungo, "Software architecture of PSET: A page segmentation evaluation toolkit," Tech. Rep. CAR-TR-955, University of Maryland, College Park, MD, September 2000. `http://www.cfar.umd.edu/~kanungo/pubs/trpset.ps`. Software is available at `http://www.cfar.umd.edu/~kanungo/software/software.html`.
30. S. Mao and T. Kanungo, "PSET: A page segmentation evaluation toolkit," in *Fourth IAPR International Workshop on Document Analysis Systems, Rio de Janeiro, Brazil*, December 2000.
31. S. Mao and T. Kanungo, "Empirical performance evaluation methodology and its application to page segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence* . (to appear).
32. S. Mao and T. Kanungo, "Empirical performance evaluation of page segmentation algorithms," in *Proceedings of SPIE Conference on Document Recognition and Retrieval, San Jose*, pp. 303–314, January 2000.