TRUEVIZ User's Manual

Version 1.0

Chang Ha Lee and Tapas Kanungo

Language and Media Processing Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742
kanungo@cfar.umd.edu

Contents

1 Overview

TrueViz (groundTRUth/metadata Editing & VIsualiZing Toolkit) is a tool for visualizing and editing groundtruth and metadata for OCR. TrueViz is developed in the Java programming language which is executable in various platforms including Windows and Unix. TrueViz reads/stores groundtruth and metadata in XML format, and reads a corresponding image stored in TIFF image file format. Multilingual text editing and display is provided by using the Unicode representation for text.

1.1 Visualization

TrueViz has two vertically splitted panels. The left panel is an image panel for displaying a document image and corresponding metadata, and the right panel is a tree view for displaying the metadata structure.

The image panel displays a document image and overlays metadata on the image. Currently, three kinds of metadata can be visualized. The bounding boxes of the entity is visualized as a polygon whose color represents the type of the entity. The logical relationship means the logical order of the document entities, and it is visualized as 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. The attribute values can be edited in the tree node and groundtruth text can be edited in the separate multilingual text editor.

1.2 Data Format

The groundtruth metadata is stored in XML file format (see Figure 1), 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. An 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.

1.2.1 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

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Page SYSTEM "Toolkit.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"/>
    <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=""/>
    <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 1: An XML example file.

line of text. A Line could be broken down into one or more Words 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.

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>
```

1.3 Special Files

- Toolkit.dtd: This file is the data type definition that governs the entire application. It is responsible for enforcing the structure of the XML used in this application. Changing Toolkit.dtd can adversely impact the application's ability to function properly. Do so with extreme caution. Note: If in the future it is necessary to alter the structure of the data for this application then this file must be updated to reflect the new structure. Also, all XML files must be updated to the new structure as well.

- Toolkit.properties: This file contains all application settings. The settings are described below:

ZoneColor, LineColor, WordColor, CharacterColor: These properties are used to specify the colors that represent Zones, Lines, Words and Characters in the application.

ArrowColor: This property specifies the color of arrows which are used to display logical relationships between entities.

- Color Values: The color values used in this property file can be expressed in two ways. The first way is to give a color name, and the second way is to give a R, G and B values where R, G and B are integers between 0 and 255, and specify the intensity of red, green and blue colors respectively. Color names recognized by TrueViz are black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow. Examples:

  ZoneColor=blue

  LineColor=0,0,255

ZoneCoordinates, LineCoordinates, WordCoordinates, CharacterCoordinates: These properties are used to determine if the coordinate information is displayed next to each vertex. If the property's value is 'true' then the information will be displayed.

ZoneArrow, LineArrow, WordArrow, CharacterArrow: These properties are used to determine if the logical relationship between entities is visualized. If the property's value is 'true' then the logical relationship will be displayed.

X_OFFSET, Y_OFFSET: Used in the construction of the rectangles that bound an ROI's vertices. Specifies the origin of each rectangle in relation to the coordinates of the vertex.

CORNER_HEIGHT, CORNER_WIDTH: Used in the construction of the rectangles that bound an ROI's vertices. Specifies the height and width of each rectangle in relation to the coordinates of the vertex.

2 Using TrueViz

2.1 Starting the Application

To run the application simply evoke the following command at a command prompt:

```
java -mx256m -jar TrueViz.jar
```

or

```
trueviz
```

Make sure that you have <TrueViz directory>/bin directory in your path before you execute this command.

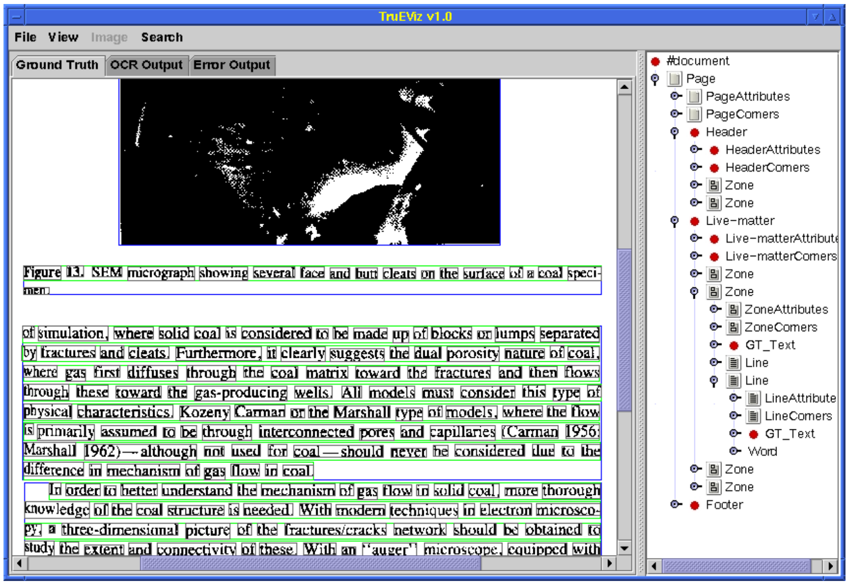


Figure 13. SEM micrograph showing several face and butt cleats on the surface of a coal specimen.

of simulation, where solid coal is considered to be made up of blocks or lumps separated by fractures and cleats. Furthermore, it clearly suggests the dual porosity nature of coal, where gas first diffuses through the coal matrix toward the fractures and then flows through these toward the gas-producing wells. All models must consider this type of physical characteristics. Kozeny Carman or the Marshall type of models, where the flow is primarily assumed to be through interconnected pores and capillaries (Carman 1956; Marshall 1962)—although not used for coal—should never be considered due to the difference in mechanism of gas flow in coal.

In order to better understand the mechanism of gas flow in solid coal, more thorough knowledge of the coal structure is needed. With modern techniques in electron microscopy, a three-dimensional picture of the fractures/cracks network should be obtained to study the extent and connectivity of these. With an "auger" microscope, equipped with

Figure 2: TrueViz Page View.

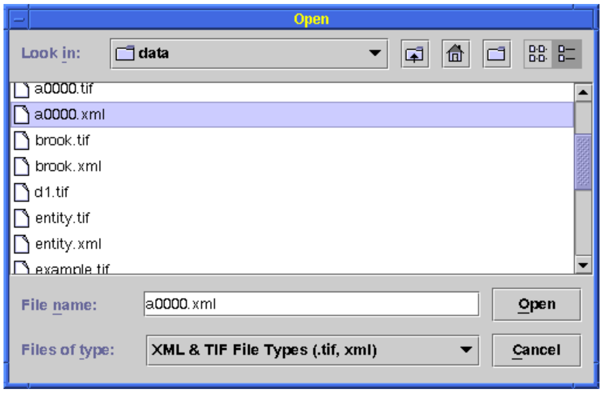
2.2 Application Layout & Operation

Once the application starts you will be presented with one window divided into two panes (see Figure 2). The left hand pane is used to view the image being edited as well as create, edit, and delete ROIs. An ROI, or region of interest, is a generic term used to describe any area of the image the user deems of interest. These areas can be classified into three categories: Zones, Lines, Words and Characters. ROIs 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.

The right hand pane contains a data (XML) viewer. This viewer displays the data in a tree structure of expandable and collapsible nodes. Only those nodes with push pin icons (that are not label 'x' or 'y') can be edited. Editing is initiated by a triple click on the node and completed by pressing the enter key.

2.2.1 Opening Files

Once you have started the application, the first thing you will need to do is open a data or image file. Clicking on the File → Open menu item on the menu bar will grant you access to the open dialog box. The box may be used to navigate your computers file structure to locate the appropriate image and data files. Clicking the open button will initiate the open process while clicking the cancel button will close the box without opening any files (see Figure 3). Both data and image files should have the same root name with different extensions. Data files have an .xml extension while the image files should have a .tif extension. For this version of the application both image and data files must be kept in the same directory. If you choose to open an image file the application will look for a corresponding data file (determined by the same file name but with a .xml extension). If a data file is found then it is loaded but if a data file does not exist one will



Look in: data

File name: a0000.xml

Files of type: XML & TIF File Types (.tif, xml)

Figure 3: A File Open Window.

be created for you by the application. If you choose to open a data file the application will look for a corresponding image file. If one is found then if will be loaded along with the data file. However, if no image file is found then you will be prompted with an error message.

2.2.2 Saving Files

When you have finished entering data for an image you will need to save the data. To save the data, access the save dialog box by clicking on the File → Save menu item located on the menu bar. The box will prompt you with the directory and name of the data file to be saved. IT IS STRONGLY RECOMMENDED THAT YOU ACCEPT THIS NAMING CONVENTION. Otherwise, the application may not find your data and thus not function correctly.

2.2.3 Changing Views

Once you've successfully opened an image or data file you will be presented with the image in the left hand pane along with any ROIs that the data file may have contained overlaid on the image. Because of the hierarchical nature of the ROIs, it is necessary to change views in order to view specific portions of that structure. To change views simply click on the View menu located on the menu bar and then select the appropriate menu item.

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 all ROIs for the image drawn from the highest level to the lowest level. This view is not editable or selectable. The Zone view shows only Zone ROIs. 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 ROI (see Figure 6). The corresponding node in the data viewer will also be selected. Similarly, the Line view shows all Line ROIs (see Figure 4 (a)), and the Word view shows all Word ROIs (see Figure 4 (b)), and the Character view shows all Character ROIs. Like the Zone view, ROIs can be selected

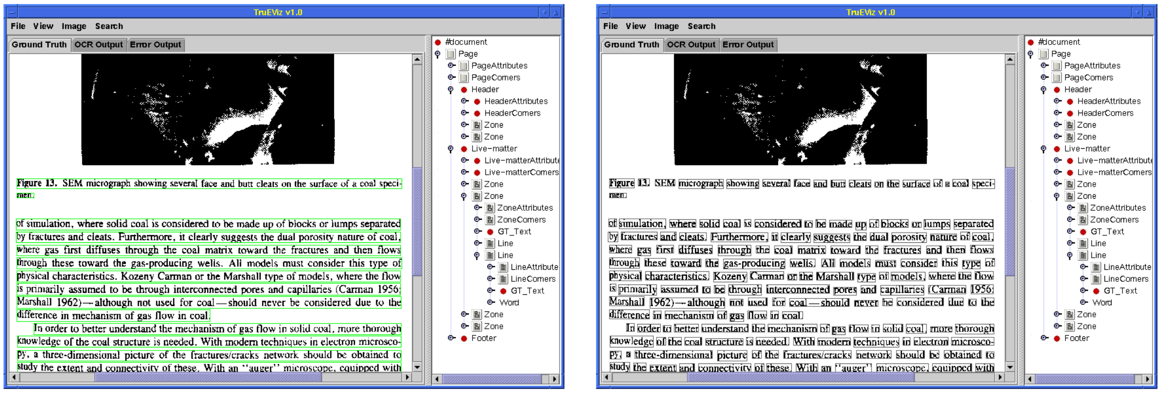


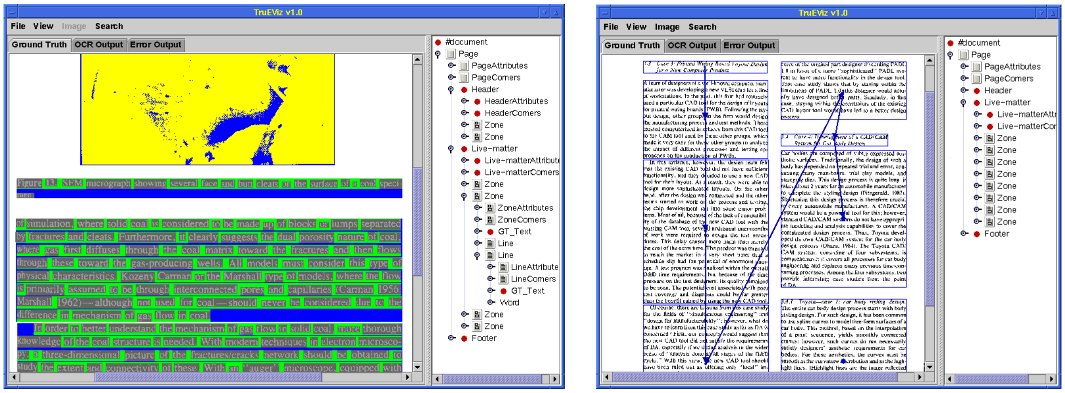
Figure 13. SEM micrograph showing several face and butt cleats on the surface of a coal specimen.

of simulation, where solid coal is considered to be made up of blocks or lumps separated by fractures and cleats. Furthermore, it clearly suggests the dual porosity nature of coal, where gas first diffuses through the coal matrix toward the fractures and then flows through these toward the gas-producing wells. All models must consider this type of physical characteristics. Kozeny Carman or the Marshall type of models, where the flow is primarily assumed to be through interconnected pores and capillaries (Carman 1956; Marshall 1962)—although not used for coal—should never be considered due to the difference in mechanism of gas flow in coal.

In order to better understand the mechanism of gas flow in solid coal, more thorough knowledge of the coal structure is needed. With modern techniques in electron microscopy, a three-dimensional picture of the fractures/cracks network should be obtained to study the extent and connectivity of these. With an "auger" microscope, equipped with

(a) Line View. (b) Word View.

Figure 4: Hierarchical Display.



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

Figure 5: View options

and the Infopanel for the active ROI is poped up in the Line, Word and Character views. Once you have a Zone selected you may change to the Line view. Only those Lines contained within the previously selected Zone will be displayed. You may interact with these Lines in the same manner as you did with the Zones. If the view is changed to the lower hierarchical level when a certain ROI is selected, the ROI selected in the previous view becomes a parent node of new ROIs created in the new view. If there is no selected ROI when the view is changed, all created ROIs become the children of the root.

There are two options for views: 'Fill Bounding Boxes' and 'Logical Relations'. These options are implemented as check boxes. If you check the 'Fill the Bounding Boxes' option, all ROIs are painted in colors corresponding to their types (see Figure 5 (a)). Otherwise, ROIs are displayed using polygonal outlines whose colors also represents their types. If you select 'Show Logical Relation' option, the logical ordering relations are visualized (see Figure 5 (b)). The logical ordering relations are visualized using arrows from an ROI to the next logical ROI. This logical order is the reading order.

2.2.4 Adding ROIs

Once you've selected the appropriate view you may add a ROI to that collection. To add a ROI simply choose the type of ROI you want from the Image, New menu. There are two types: Rectangles and Polygons. To create a rectangular ROI, select rectangle from the new menu. Your cursor will become a cross. Move the cross to where you would like the ROI to begin then click and hold the left mouse button. Drag the mouse to properly size the ROI and then release the left mouse button to complete the ROI creation. To create a polygon ROI select polygon from the new menu. Your cursor will become a cross. Move the cross to where you would like the ROI to begin then click left mouse button. You will notice a square (which represents a vertex) appears and a line will be drawn from that square to wherever you move your cursor. Continue clicking the left mouse button to fix as many vertices as necessary. To complete the polygon click on the original vertex you fixed.

2.2.5 Editing ROIs

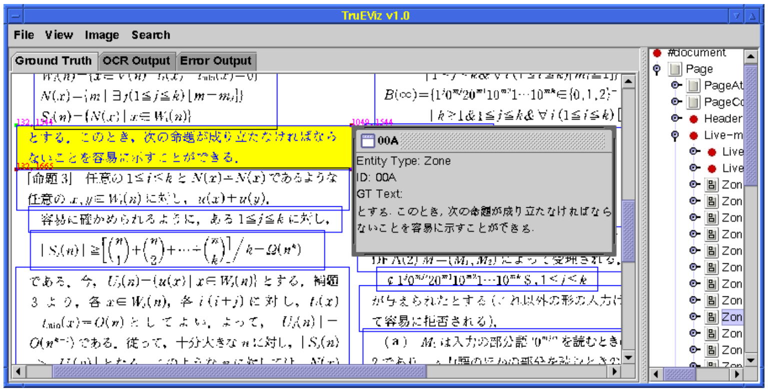
Once you've selected an ROI there are several thing you can do to it. First you should be aware of the active vertex. All the ROI's vertices will have squares (handles) located on top of them. The active vertex is the handle painted in green. This vertex is the one around which editing centers. To change the active vertex simply hold down the control key and click on the appropriate handle. Once you have the proper vertex selected you may move the entire ROI as follows. First, holding down the shift key. Then click and hold the left mouse button and move the mouse the new location. Releasing the shift key and the mouse button finishes the move operation. You can also skew the ROI by holding down the alt key and clicking and holding the left mouse button. Now when you move the mouse the active vertex will follow your mouse. Releasing the alt key and mouse button finishes the move of the active vertex. Finally you can rotate an ROI by pressing either the comma or period keys. The comma key rotates the ROI clockwise around the active vertex and the period key rotates the ROI counter clockwise around the active vertex. NOTE: It is extremely important that you click on the ROI after completing rotation. This will permanently fix the new ROI coordinates. If you bypass this step the ROI's coordinates may NOT get fixed and some data may be lost.

2.2.6 Deleting ROIs

Once you have selected an ROI you need only to press the delete key or select Delete ROI from the menu bar. WARNING: This application does not have an UNDO feature. Therefore, delete ROIs with extreme caution as deleting an ROI not only deletes the ROI but also any children the ROI may have. For example deleting a Line will also delete any Characters associated with that Line.

2.2.7 Editing Groundtruth

Groundtruth metadata can be edited in two ways: graphical editing and text editing. Bounding boxes and logical ordering relations can be edited graphically. Editing bound-



00A

Entity Type: Zone

ID: 00A

GT Text:

とする．このとき，次の命題が成り立たなければならないことを容易に示すことができる．

Figure 6: Infopanel and Multilingual Display.

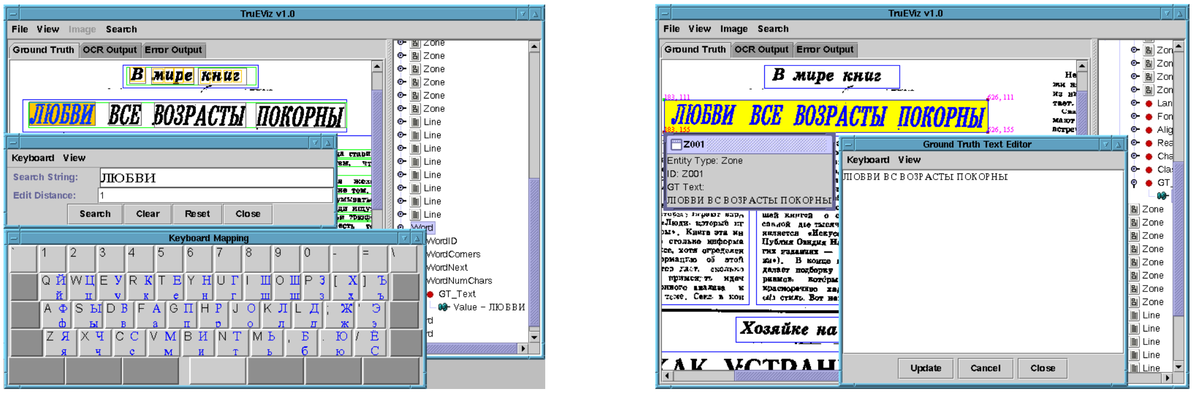
ing boxes is described in the Editing ROIs section. To delete a certain logical relation, select the previous ROI and select Image → Delete Logical Relation. To set a new logical relation, select the previous ROI and select Image → Set Next Entity, then click on the next ROI. Attribute values can be edited within the node of the tree view by a triple click on the node. Because the groundtruth text may contain multilingual text, the groundtruth text is edited in the separate multilingual text editor. The multilingual text editor is evoked by a double click on the node. To change the input method, select Keyboard menu in the editor and choose the input method you want. In case you are not familiar with the keyboard mapping, you can bring up the keyboard mapping display by selecting View → View Keyboard menu.

2.2.8 Multilingual Input Method

TrueViz provides a multilingual input system. Besides the default 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 7). For example, TrueViz supports multilingual text input in the search window for multilingual search. To change the input method, select Keyboard menu and choose the input method you want. In the case you are not familiar with the keyboard mapping, you can bring up the keyboard mapping display by selecting View → View Keyboard menu. In addition to the keyboard input system, TrueViz provides a Unicode character input using a code table (see Figure 8). By selecting Keyboard → Unicode Table menu, any unicode character can be selected and inserted into a text.

2.2.9 Search

TrueViz provides a multilingual approximate search functionality. To search a certain string in the groundtruth text of the document, select Search → Find String menu item. A search string and edit distance can be specified in the search window. TrueViz provides multilingual input methods for a search string. To change the input method, select Keyboard menu in the search window and choose the input method you want. In



Search String: ЛЮБВИ

Edit Distance: 1

GT_Text

Value – ЛЮБВИ

Z001

Entity Type: Zone

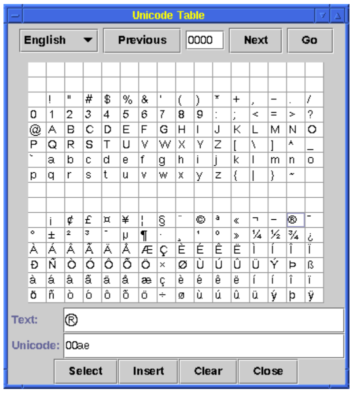
ID: Z001

GT Text:

ЛЮБВИ ВСЕ ВОЗРАСТЫ ПОКОРНЫ

(a) Russian Input Method in Search. (b) Russian Input Method in Groundtruth Editor.

Figure 7: Multilingual Input Method

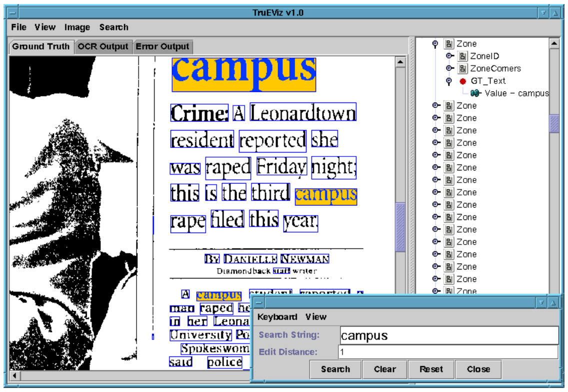


	!	"	#	$	%	&	'	(	)	*	+	,	−	.	/
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	−	®	¯
°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Text: ®

Unicode: 00ae

Figure 8: Unicode Table.



campus

Crime: A Leonardtown resident reported she was raped Friday night; this is the third campus rape filed this year.

By Danielle Newman

Diamondback staff writer

Search String: campus

Edit Distance: 1

Figure 9: Search.

case you are not familiar with the keyboard mapping, you can bring up the keyboard mapping display by selecting View → View Keyboard menu. The edit distance is the minimum number of substitutions, insertions and deletions required to transform one string into the other. The maximum edit distance allowed during the search can be specified. After the search is finished, all ROIs containing the search string with the specified edit distance are highlighted (see Figure 9).

2.2.10 Exiting the Application

To quit the application, simply click on the File → Exit menu item. Warning: This will close the application without prompting the user to save. Be sure to save all data before exiting.

Acknowledgements

Authors would like to thank Glenn van Doren of the Department of Defense for valuable input. This project was funded in part by the Department of Defense, the Defense Advanced Research Projects Agency, Lockheed Martin, and the National Science Foundation. We would also like to thank Jeffrey Czorapinski and Ivan Bella of Lockheed for discussion and Song Mao of University of Maryland for discussion and testing.