ShotVis: Smartphone-based Visualization of OCR Information from Images

BIAO ZHU, Zhejiang University
HONGXIN ZHANG, Zhejiang University
WEI CHEN, Zhejiang University
FENG XIA, Dalian University of Technology
ROSS MACIEJEWSKI, Arizona State University

While visualization has been widely used as a data presentation tool in both desktop and mobile devices, the rapid visualization of information from images is still underexplored. In this work, we present a smartphone image acquisition and visualization approach for text-based data. Our prototype, ShotVis, takes images of text captured from mobile devices and extracts information for visualization. First, scattered characters in the text are processed and interactively reformulated to be stored as structured data (i.e., tables of numbers, lists of words, sentences). From there, ShotVis allows users to interactively bind visual forms to the underlying data and produce visualizations of the selected forms through touch-based interactions. In this manner, ShotVis can quickly summarize text from images into word clouds, scatterplots, and various other visualizations all through a simple click of the camera. In this way, ShotVis facilitates the interactive exploration of text data captured via cameras in smartphone devices. To demonstrate our prototype, several case studies are presented along with one user study to demonstrate the effectiveness of our approach.

Categories and Subject Descriptors: H.5.2 [**User Interfaces**]: Interaction styles

General Terms: Algorithms, Human Factors

Additional Key Words and Phrases: Cyber-physical interaction, Data Visualization, Touch-based Interface, Interaction, Smartphone, Wearable Computing

1. INTRODUCTION

Imagine a menu at McDonalds, labels on grocery store shelves, data tables in text books. Underlying each of these items is a massive quantity of data that can be captured and explored. While visualization is perhaps one of the quickest means of comparing the price of twenty cans of tomato soup, or plotting trends from data tables, visualization only truly lends itself to digitized information that can have structure imposed upon it. The fact is, it is difficult to transform information found in the real world into digital form. While tools for turning McDonald's menus into a data file exist, solutions typically require professional hardware and toolkits from computer vision [Szeliski 2010], video processing [Wang et al. 2001], text analysis [Miner et al. 2012], and signal processing [Lyons 2010]. These tools are simply not readily available to non-professional/casual users. As such, it is uncommon to find ubiquitous visualization tools for daily-life information analysis. However, the global adoption of smartphones has led to the rise of ubiquitous computing [Weiser

Author's addresses: Biao Zhu, Hongxin Zhang and Wei Chen, State Key Lab of CAD&CG, Zhejiang University; Feng Xia, Dalian University of Technology; Ross Maciejewski, Arizona State University;
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2015 ACM 1551-6857/2015/03-ART39 $15.00
DOI:http://dx.doi.org/10.1145/0000000.0000000

1999; Oulasvirta et al. 2012]. What smartphones provide is a handheld camera and universal means of capturing data through images or video. With access to high-resolution cameras and high-speed wireless networks, smartphones are naturally suited for the capturing, processing and analysis of information, and are being increasingly used as multimedia computation and illustration platforms [Munzner 2014].

While the ubiquity of smartphones can provide a means of capturing and processing data, there is a distinct need for *understanding* data. Traditionally, visualization is one of the most important, and commonly used, methods of generating insight into large scale data. Particularly for textual or tabular data, visualization is used to create summaries that take advantage of the power of the human perception [Roberts et al. 2014]. Even though visualization has been widely used as a data presentation tool in both desktop and mobile devices, the rapid visualization of information from visual media (e.g., images, videos) is still underexplored. In addition, most previous works focus on the visualization of datasets that are recorded as data tables or formatted data files. To the best of our knowledge, there is little work dedicated to visualizing information contained in images and videos.

The presented work is motivated by the fact that smartphones are the most generally available computational devices. By utilizing the ubiquity of smartphones and the ability of smartphones to capture and process data, our goal is to enable visualization of information that is not readily available as a data file. In doing so, we will provide user with a means of generating insight into non-traditional data sources. Our work focuses on the integration of a suite of data processing techniques, including cameras, touch-based interfaces and optical character recognition (OCR) [Microsoft 2014] within an interactive data acquisition and visualization pipeline. The unique feature of this pipeline is that it retranslates text information contained in images into visual forms, thereby providing users with a viable means of generating insight from large quantities of (potentially) non-digitized data.

While the data capture and processing portion of the pipeline is challenging, it is important to note the inherent challenges that visualization faces on smartphones. These devices have limited screen space and computing power. Furthermore, interaction traditionally found through a keyboard and mouse is replaced with touch and gesture interactions. Our goal is to develop an interactive system for data capture, processing and analysis that is both visually expressive and intuitive to use.

In this paper, we describe a novel mobile visualization scheme that builds upon the latest OCR, touch interaction and visual design techniques, and leads to a new portable visual design tool. Our contributions include:

— a new visual transformation mode that reformulates camera-captured images into expressive visual representations;
— an efficient touch-based interaction scheme that allows for direct manipulation and structuring of scattered characters into visualizable data forms;
— a smartphone-based visualization system that supports the recognition, structuring and design of visualizations based on camera-captured images.

Previous work has primarily focused on designing visualizations that are appropriate for lower screen resolutions and proper interaction techniques in the context of mobile devices, e.g. [Chittaro 2006; Burigat et al. 2006]. Specific layout and interaction principles for mobile devices are described in Yoo et al. [Yoo and Cheon 2006], and design principles from past work were used during our prototype development. Our approach advances work in this area by designing an easy-to-use means of data acquisition and visualization for mobile devices. While the popular usage (e.g., indoor navigation) of camera-captured images is for photo sharing, our approach allows

users to seamlessly turn images that contain text into visualizations. Examples include understanding a large textual paragraph with tag-cloud visualizations, reading data form hand-written charts, and comparing multiple product specifications with graphs, as demonstrated in the result section. Most similar to our proposed work is ReVison [Savva et al. 2011], which seeks to scan and automatically convert existing charts (data visualizations) back to a tabular data format. Once the data is acquired in ReVision, it can then be transformed into new visualizations using different visual designs, for example, changing pie charts to bar charts. The pipeline in ReVision specifically focuses on extracting information from existing visualizations using advanced pattern recognition and data mining techniques. Our works differs from ReVision in that we focus specifically on characters rather than rendered charts. Furthermore, our work leverages smartphones and user interactions to make it a practical solution for novices and experts alike.

2. RELATED WORK

As previously stated, the most similar work to our proposed pipeline is the ReVision system by Savva et al. [Savva et al. 2011]. ReVison was developed to take images of existing charts, turn these charts back into the original structured data set and then allow users to turn them into new visualizations. In this manner, data that was only previously available as bitmap images can now be extracted, edited and revisualized. Unfortunately, the data extraction rate of bitmap-image charts is relatively low even with state-of-the-art techniques. In some cases this is caused by poor image quality, while in other cases it is caused by fidelity issues such as tiny sectors within a pie chart or low resolution axis labels on a grid. While the goal of ReVision was to extract and remap data from charts, our proposed pipeline is to take images of textual data found in the physical world, digitize the data and then allow users to visualize the captured data. While tools such as Office Lens Office Lens [Microsoft 2014], provide text recognition for mobile devices, our goal is to expand from text recognition to visualization. Although our pipeline shares similar factors to that of OfficeLens [Microsoft 2014] with regards to data capture and recognition, ShotVis is unique in that our pipeline encompasses an all-in-one targeted visualization solution. Moreover, our approach is focused on the data manipulation and visual design aspects which are key issues for visualization authoring directly on smartphones. While OfficeLens can be employed as an OCR backend of our approach, its functionality would only serve part of the proposed pipeline. Thus, our paper presents a pipeline for capturing data in the physical world, transforming this data into digital information and then providing multiple visualization options for viewing the newly captured data.

It is important to note that our work is not the first to propose using smartphones for data capture and visualization. Chittaro et.al. [Chittaro 2006; Burigat et al. 2006] discussed different aspects of visualizing information on mobile devices, such as how to visualize locations of off-screen objects. Yoo et.al. [Yoo and Cheon 2006] proposed different visualization methods for different data types and found that sequential layouts are suitable for data with less relational information while radial layouts are more suitable for hierarchical information. Hao et al.[Hao and Zhang 2007] proposed an interface for hierarchical information, which takes advantage of connection and enclosure approaches within the limits of screen resolution on mobile devices. Work by Kim et al. [Kim et al. 2008] and Razip et al. [Razip et al. 2014] explore using mobile devices for improved situational awareness, and several research projects (e.g. [Zhou et al. 2006; Lamberti and Sanna 2007]) explored 3D visualizations for mobile devices. However, all of the aforementioned works utilize data that is preprocessed and store either on the mobile phone or through servers available to the

client. In these cases, the visualizations are designed based on a known data structure, format, and visualization type. In ShotVis, we want to enable users to capture more free-form data; however, since the data type being captured is unknown, our visualization needs to handle data capture, data manipulation and visualization of unstructured data. To the best of our knowledge, our work is the first time a full data capture to visualization pipeline has been proposed for a mobile device.

With regards to data capture, smartphones are equipped with a large number of sensors (e.g., microphones, motion sensors, cameras, etc.) which can be used to capture information in a user's environment. These devices can be further augmented with external sensors, and their portable form factor and ubiquity has led many researchers to begin exploring how one can utilize the data acquisition abilities of a smartphone to improve peoples' day-to-day lifestyle. For example, Buttussi et al. [Buttussi and Chittaro 2008] and Kanjo et al. [Kanjo et al. 2008] both utilized mobile devices to collect and visualize geolocation and movement data respectively. Macias et al. [MaciasMacías et al. 2012; Macías et al. 2011] tags videos with data collected by sensors embedded in the phones. Andrey Vasilev et al. [Vasilev et al. 2012] visualize context captured in smart space with HiveMind. Girod et al. [Girod et al. 2011] developed a tool where a picture snapped with a handheld device becomes a search query, allowing things like comparison shopping.

While previous researchers have used mobile sensors for the capture and visualization of data, our work moves past the type of passive data collection as seen in Buttussi et al. [Buttussi and Chittaro 2008] and Kanjo et al. [Kanjo et al. 2008] and brings the user into the data capture process in a manner similar to Girod et al. [Girod et al. 2011]. The difference here is that our data capture is not intended for search queries, but it is intended for sense making and knowledge acquisition, taking free-form data found in the physical world and turning this into digital information that can be visualized. In order to capture free-form data (specifically text data), we utilize optical character recognition tools (OCR) [Wikipedia 2014], which is used for the conversion from image text to machine-encoded text. These tools are a key part of our proposed pipeline in that they transfer image blocks containing scattered characters into numbers and words. Once the data is captured, the data then needs to be organized and structured for visualization purposes. Given that the data we are proposing to capture is free-form, we need to incorporate data wrangling techniques for labeling and manipulating the data to make it appropriate for visualization. Data wrangling is the process of manually converting data from one form into another format that allows for more convenient consumption of the data with automatic tools.

Recent work in the visualization community has explored the development of tools for aiding the data wrangling process. Specifically, Wrangler [Kandel et al. 2011] was developed to combine the direct manipulation of data with automatic inference of relevant data transforms, thus relieving some of the manual burden from the user and making data wrangling more efficient and effective. However, Wrangler focuses on structured but incomplete or faulty data. In our application, the data encountered in daily life is typically unstructured (and often incomplete). To manipulate such unstructured data, the traditional approach is to write scripts in Python, Perl and R to transform the data, or to manually edit the data with tools like Microsoft Excel. However, such massive editing and script writing is not transferable to mobile devices. As such, our work explores designing data wrangling operations specific for smartphones to enable intuitive and efficient data manipulation on mobile devices.

Different from the "mouse-and-keyboard" interaction in desktop PC environments, interaction design on smartphones is crucial for visualizing data on small screens. A natural solution is to support sketch gestures for smartphone applications instead of a button-based interface. Furthermore, sketching often conveys information that can be

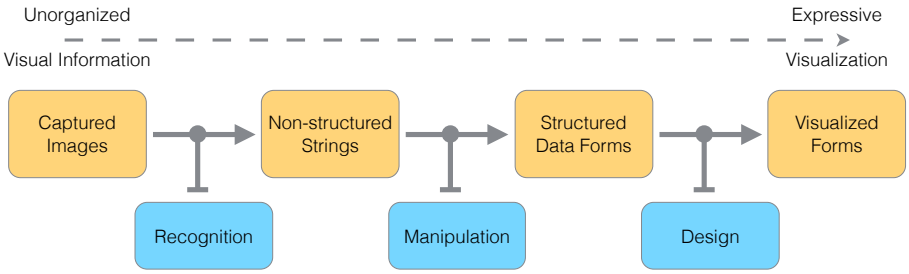


Fig. 1. The ShotVis Pipeline.

hard to explain using text or button inputs, while at the same time sketching does not require a tremendous amount of effort from the user. As human-computer interaction moves towards higher level languages, sketching will certainly continue to have a place in mobile applications and visualization. Rather than employing complex sketch shape classifications in the command recognition stage [Schneider and Tuytelaars 2014], we apply a relatively simple way to recognize users' input [Wobbrock et al. 2007] using a 2D single stroke. [Girod et al. 2011]

3. SHOTVIS

In this section, we describe our pipeline for capturing data in the physical world, transforming this data into digital information and then providing multiple visualization options for viewing the newly captured data. Our approach is applicable for scenarios where textual information (recognizable characters or numbers) is physically presented, for example in a booklet, a poster or a hand-written whiteboard. While such data is readily available for human consumption, our goal is to automatically synthesize this data and transform this into visualizations. In this way we hope to enable users to quickly gain insight into large amount of physical data. Our approach combines automatic text recognition, data manipulation/wrangling and visualization design with requirements of a modest amount of user interactions. Fig 1 presents the ShotVis pipeline together with the flow of user interaction.

First, a user takes a photograph of text found in the physical world with the smartphone. Automatic image alignment and character recognition are performed on the photo. Then, the recognized characters are interactively organized and manipulated into structured data tables or forms. An initial organization is suggested, and, subsequently, the user manually maps the data into their desired visual forms by selecting visual transformations and encodings.

3.1. Data Recognition

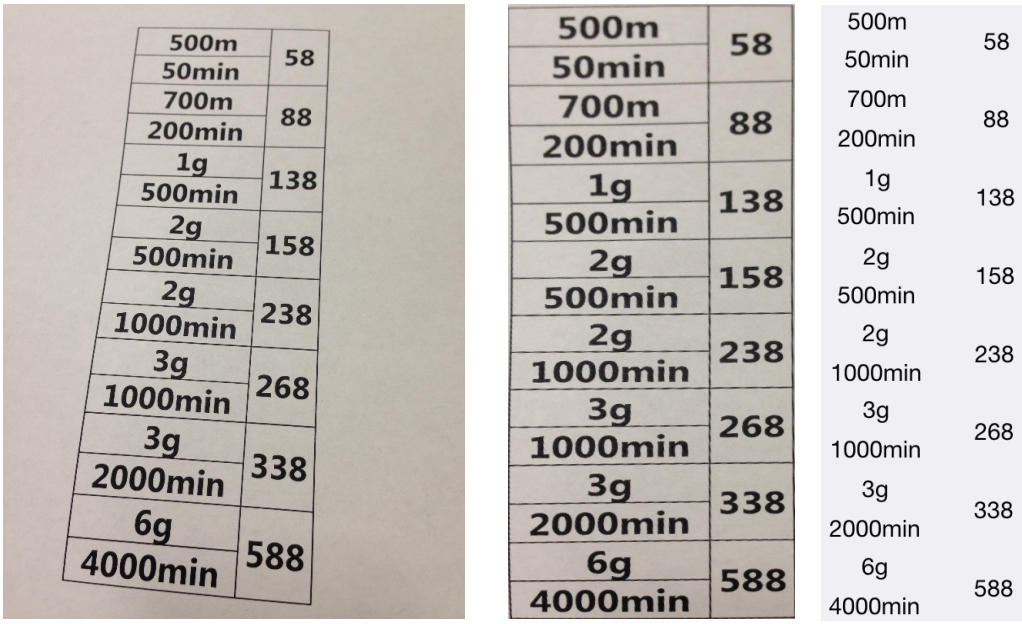
Suppose that a user captures one image $\mathcal{I}$ with a smartphone, and the image contains information (hand-written characters or numbers which may be freely placed or organized as tables, forms, or lists) that that the user wants to visualize. For the sake of simplicity, we denote such images as raw data $\mathcal{R}$. Conventionally, visualization is applied to datasets that are stored as data tables or data forms. If the targeted data is displayed in physical media, such as books or slides, a data scanning process must be performed to digitize the data. The process includes three steps:

— First, a captured image $\mathcal{I}$ (Figure 2 (a)) is cleaned and registered with a state-of-the-art image warping technique [Lee et al. 2014]. Figure 2 (b) shows the warped result of Figure 2 (a).

— Second, an OCR process is performed on the calibrated image, resulting in a list of N scattered characters or strings $S_i (i = 1, 2, ..., N)$ (Figure 2 (c)). An additional check is then performed to classify the data into a category, such as tabular, paragraph, unrecognized image, etc.
— Third, for each S_i, an additional attribute set A_i is automatically recognized for the purpose of preserving consistency between the raw data and the final visualization. The attribute set A_i is actually a triple $< f_i, p_i, l_i >$ which consists of a font color of the physical characters f_i, a location center coordinate p_i of string S_i, and a set of category labels l_i. Currently, several content category label are supported in our implementation including *text*, *numbers* and *geographical locations*. However, the definition of the category labels can be dynamically extended for future data acquisition and visualization purposes.

Finally, the raw data $\mathcal{R} = \{(S_i, A_i) | i = 1, 2, ..., N\}$ is generated.

The output of OCR is unstructured data which consists of a set of strings S_i with corresponding 2D coordinates $p_i = (x_i, y_i)$. If the data $\mathcal{R}$ is categorized as tabular (as in Figure 2 (c)), only a single preprocessing step needs to be performed to construct a table-like structure from the image. To achieve this, we utilize *integral-projection* row-column relationships for all S_i. We project all positions p_i to the X- and Y-axis. After this transformation, coordinates are clustered into several groups and a simple median thresholding is applied to distinguish columns and rows.



(a) Captured image (b) Alignment (c) OCR result

Fig. 2. Illustrating the process of data scanning.

3.2. Data Manipulation

As shown in Figure 3(a-b), raw textual data is directly generated from an image. However, the generated data only consists of characters and their corresponding locations within the image. Without other contextual information, these data are not

ready for visualization. The resultant data are unstructured, in other words, there are no rows, columns, or subtables. Along with the missing structure problems, the generated data may also contain errors due to poor character recognition. Thus, in order to visualize the data, users must clean, select and label dimensions from the raw data through a series of interactions.

3.2.1. Gestures. The interaction for data manipulation consists of two simple steps:

(1) the user selects data via sketching gestures
(2) the user assigns data labels to the set of selected data.

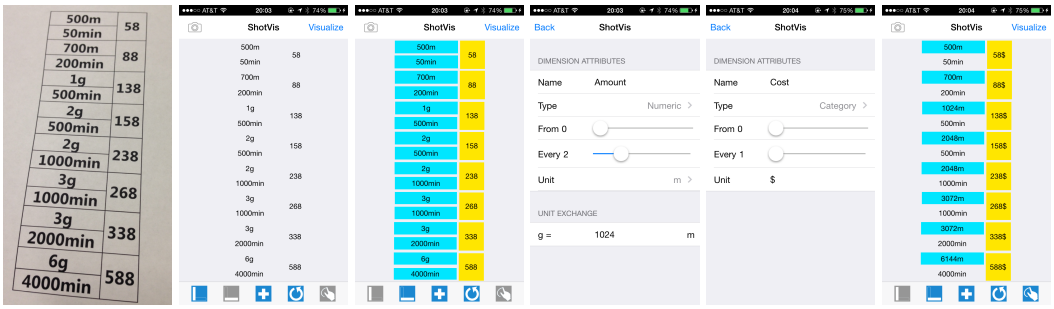
ShotVis recognizes several gestures which can be sketched by the user and automatically manipulates the data according to the supported gestures. Figure Figure 4 shows the four sketching gestures that can be used to label and assign data variables to axes. To create a chart (bar chart, tally chart, line chart, and scatterplot) , the user will create an arrow sketch; the clear operation is invoked with a circle gesture, and; adding a data group is invoked with a one touch "+" sketch.

Sketching operations are processed via a lightweight gesture recognition library, $1 Unistroke Recognizer [Wobbrock et al. 2007]. This library is a 2-D single-stroke recognizer designed for the rapid prototyping of gesture-based user interfaces. To fully utilize the touch interface of a smartphone, a two-finger touch gesture is applied as an on/off switch to change between the gesture command mode and the true editing-and-selection mode. Alternatively, a row of five buttons has been added to the bottom of the screen. Each button, when activated, corresponds to one of the five sketching gestures (including the switch gesture), and each button is rendered with the corresponding gesture symbol for ease of use.

These operations have been specifically designed to help structure the data for visualization. Typically, for visualization, each data attribute describes a single aspect of a record, and multiple records form a data set. To visualize such a data set, we need to map each data attribute to a separate visual channel. For example, mapping attribute A to the x-axis and attribute B to the Y-axis creates scatter plot.

3.2.2. Labeling. On the bottom of Figure 3(b) there exists two visual channel buttons corresponding to the x and y axes. To select and map an attribute to a data axis, users need to first tap on either the axes-Y-button or axes-X-button. After that, users can then select data by tapping on or swiping over each data entry. Once the data for an attribute is selected, users can then tap on the corresponding visual channel button again to further edit the data attribute. The editing interface is shown in Figure 3(d)(e). This interface allows users to perform editing operations including assigning a name/type/unit to the attribute.

As part of data labeling, users may also wish to derive data characteristics from the data set. For example, Figure 3(a) contains information about the *price*, *net flow* and *call time* of each package. However, a user may wish to visualize the unit cost of *net flow* or *call time*. This would require algebraic manipulation of the data. ShotVis enables users to perform algebraic operations with existing data, as illustrated in Figure 5. First, in order to get the unit cost dimension, two dimensions, *net flow* and *call time*, are selected(Figure 5(a)). Then a long-tap interaction is performed upon the blank area to activate the algebraic operation. Next, in Figure 5(b), the name of each selected dimension and the available algebraic operators are all shown as draggable labels. Users can drag these labels into the white area to construct an algebraic expression and derive new data dimensions, Figure 5(c). Finally, after a user confirms the algebraic operation, the new dimension will be added to the original data form, Figure 5(d).



(a) (b) (c) (d) (e) (f)

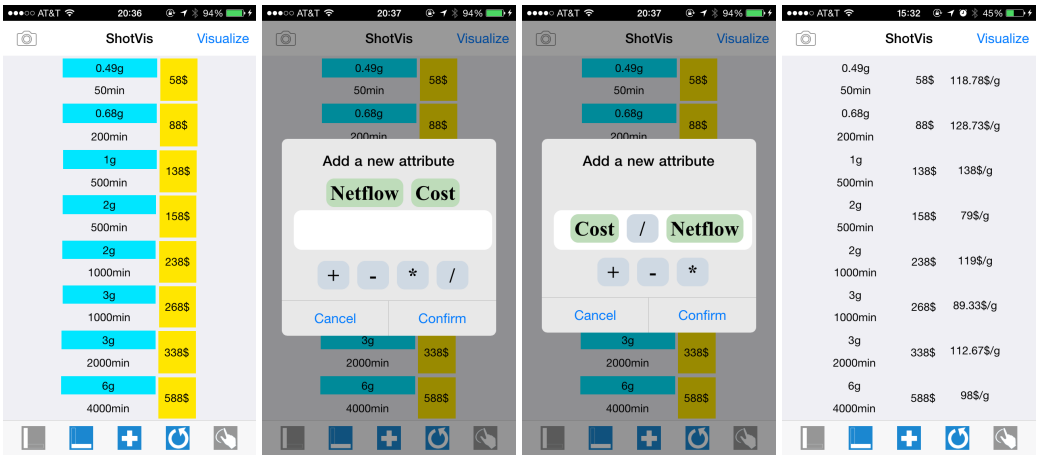
Fig. 3. (a) A camera image of the physical print data. (b) Optical character recognition has been performed. (c) Data has been selected by the user. (d-e) The data is then mapped to data attributes via the user interface. (f) The final mapped data table is then viewed by the user.

(a) (b) (c) (d)

Fig. 4. Sketch gestures for data manipulation. (a) select data for X axis. (b) select data for Y axis. (c) clear data selection. (d) add a data group.

3.2.3. Cleaning. While gestures allow users to label and manipulate data, it is possible that the OCR algorithms used will cause the scanned data to contain errors. As such, it is often necessary to clean the data. To improve data cleaning, the captured data is labeled using a specific color coding for all recognized string blocks. Blocks filled with red indicate that there is likely to be an unrecognized character string in that element. Such data can be corrected in the value editing mode. While users can manually adjust the data, this process would be burdensome, and we have automated many data cleaning tasks in order to make ShotVis easy to use.

Our first automated option involves *automatic data matching* which allows users to make efficient data selections and extractions by modifying two parameters, *From* and *Every*. This option is necessary when data attributes are co-located in a column. For example, in Figure 3(a), two dimensions, *net flow* and *call time*, are co-located in the same column. Conventionally, instead of swiping, the user would need to select, by tapping each item, and then label each data value as part of a data dimension. This process, as described, can be very slow and inaccurate. However, adjusting *From* and *Every* can make this process quite efficient. First, the user selects the whole column by swiping. Then, by setting *From* to 0 and *Every* to 2, the $1st, 3rd, 5th \ldots$ items, which consist of the *net flow* dimension are grouped, and setting *From* to 1 and *Every* to 2 will select the $2nd, 4th, 6th \ldots$ items, which consist of the *call time* dimension. Thus, by using our automatic data matching tool, the *ith* items can be selected, where $i \in$ $\mathit{From} + n \times \mathit{Every}, n = (0, 1, 2 \ldots)$. Furthermore, ShotVis is designed to automatically recommend the value of *From* and *Every*. Again, taking Figure 3 as an example, if we choose the right column first and then the left column, ShotVis will detect that the number of rows in the left column is twice the number of rows in the right column. ShotVis will then recommend *Every* to be 2 and leave *From* to 0, which automatically selects the *net flow* dimension.



(a) (b) (c) (d)

Fig. 5. Algebraic operation illustration: (a) two dimensions are selected. (b) initial interface. (c) after user interaction. (d) a new dimension, $Unit\ cost\ =\ Cost/Netflow$, is generated.

Our next automated option includes *content-aware editing*. When the data table contains information about an attribute, but the data is recorded differently (part of the data is metric units and part is British units), we need to make unit conversions in order to have an appropriate visualization. In this case, the user must choose the default unit and define the conversion ratio. We have predefined the most common units and their conversion ratios, thus making this operation automatic in many cases.

Next, data extracted needs an *automatic data type assignment*. In ShotVis, we define three data types, category, numeric and geography. Different combinations of data types lead to different types of visualization. For example, a combination of categorical and numeric attributes will result in a bar chart or a line chart, while a combination of purely numeric attributes can result in a scatter-plot. Furthermore, when geographical attributes are detected, maps can be produced. Attribute data starting with a letter (A-Z character) are automatically labeled as a categorical attribute, except when the data are exact geography locations (e.g. names of cities, countries and continents), in that case the data will be labeled as a geography attribute. Attribute data starting with a number (0-9) are automatically labeled as a numeric attribute. Note that the type assigned by ShotVis is a recommendation which the user can change in the case of an error.

Since multiple attributes may be captured, we have developed tools for *automatic data group recommendations*. In ShotVis, we employ an ”add” operation to realize this functionality. Each time a user finishes a selection from a data group, the user can press the ”add” button to confirm this selection and start a new one. With the knowledge that data groups usually appear one by one in regular tabulation (e.g. the data shown in Figure 7(a)), we can automatically recommend the selection of the next data group after user finishes the selection of a data group.

3.3. Data Visualization Design

Once data is cleaned, labeled and selected, the next goal is to visualize the data. After data processing, an initial visualization is automatically generated according to the combination of data types. However, users all have personal preferences and may want to make adjustments to the automatically generated visualization.

When creating an information visualization solution, a major challenge is selecting the best visual option to represent the data. There are two main approaches for designing a visualization, namely exploratory and explanatory visualization. Choosing one depends on the context of use, the preset goal, the intention of the user, and the potential audience.

Exploratory visualizations are used to discover patterns, trends, or sub-problems in a captured data set. They are useful for performing analysis on data sets without having a detailed understanding of the data content. Explanatory visualizations are used to transmit information or a point of view from the user (in this case the visualization designer) to an audience. Unlike exploratory visualizations, explanatory visualizations are employed when the user knows the story behind the data and would like to communicate it to an audience through visualization.

In order to enable customizable visualizations, ShotVis allows users to make interactive and content-aware revisions to a visualization. For example, users can try different visualization methods for the same data as shown in Figure 6. Users can also change the title and the string colors as demonstrated in Figure 6(c-d). For the colors, ShotVis includes palettes designed for perceptually discriminable colors [Harrower and Brewer 2003] and also for the colors collected from the original bitmap-image data, as discussed in Section 3.1. While color schemes and line styles can drastically alter how a visualization is perceived and received by an audience, the focus on our work has been on the visualization pipeline of ShotVis. Future work will focus on automatically recommending appropriate color schemes and line styles based on data types (sequential data should use a sequential color scheme, etc.).

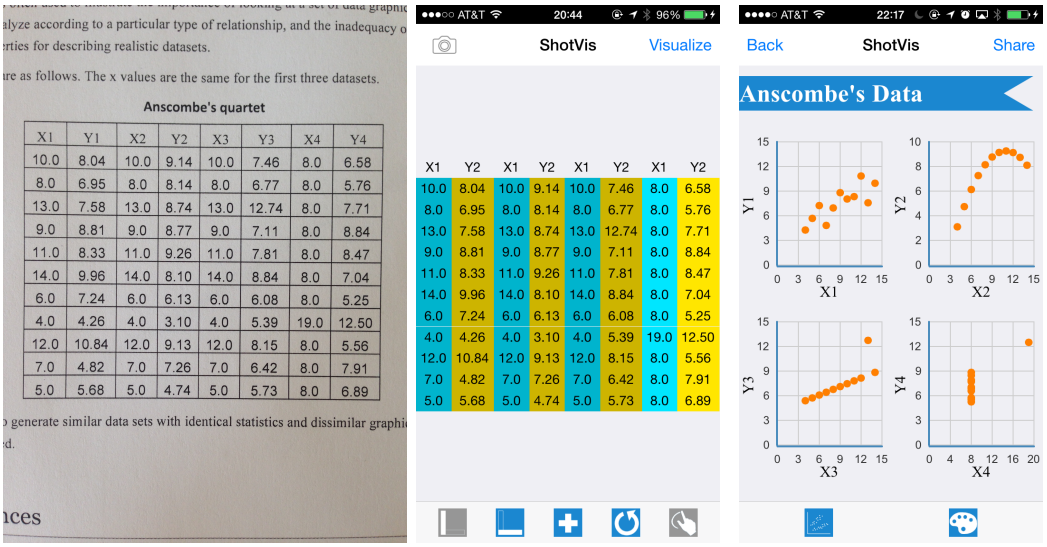
Fig. 6. (a) the initial visualization. (b) changed to line chart. (c) changed to a new palette. (d) title modified.

4. CASE STUDIES

ShotVis aims to enable the visualization of printed textual data. Traditionally, when data is not available in digital form, we need to input it into a computer first, which can be very time consuming. With ShotVis, users only need to take a photo, then the data can be seamlessly translated into a digital datafile. In this section, several examples of how ShotVis works are presented to demonstrate its effectiveness.

4.1. ANSCOMBE'S DATA

In this case, we take Anscombe's Data (see Figure 7(a)) as an example case to demonstrate ShotVis. This data contains four datasets, the mean, variance, correlation and linear regression of each dataset are nearly identical; however, their distributions in a Cartesian coordinate system are quite different. Here, visualization becomes a very useful tool for revealing such differences and helping people understand the data better. With ShotVis, we can visualize Anscombe's data simply by taking a picture of our textbook as shown in Figure 7(a). Then four groups of data are selected with the "+" operation(Figure 7(b), which is discussed in Section 3.2.1. Finally, the four data groups are visualized side by side. As we can see in Figure 7(c), the data distributions are clearly revealed by visualization where simply looking at the chart alone would be insufficient.



(a) (b) (c)

Fig. 7. (a) Anscombe's Data. (b) Choose multiple data groups in ShotVis. (c) Anscombe's Data visualized with ShotVis.

4.2. PAPER DATA

Many people read every day, they read newspapers, articles and books. However, reading is time consuming. Even for experienced researchers, it takes several minutes for them to summarize what an article is about. Word clouds are an effective visualization for summarizing the most frequent terms in an article. Word clouds first determine simple statistics about the frequency of the words in an article and then adjust words' size and position according to their frequency.

However, in the real world we often only have access to reading material in printed form, not digital. Creating word clouds from print data would require a user to transcribe the print material back into digital form. However, with ShotVis, the user can skip the transcription step entirely, replacing it with the simple act of photographing the physical document. As an example, we took a photo of the (1st draft) abstract and introduction of this paper with ShotVis (Figure 8(a)(b)) and visualized it using a word cloud(Figure 8(c-e)). From the results of the word cloud, we

can get an overview of the keywords of this paper, including “ShotVis”, “media”, and “Visual”.

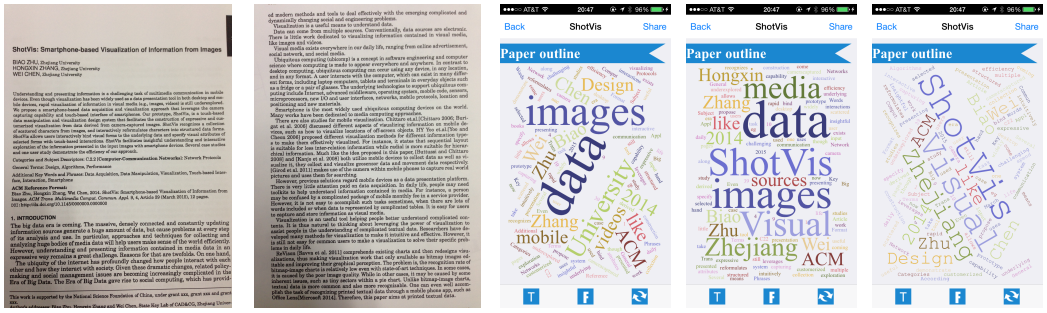


Fig. 8. (a) and (b) photos of the abstract and introduction in this paper. (c) word cloud visualization with orientation modification in (d) and with font modification in (e).

4.3. MAP DATA

The target of ShotVis is to assist users in visualizing all kinds of printed information. This requires ShotVis to provide multiple kinds of visualizations. Here we demonstrate the visualization of geographic information (Figure 9(a)). ShotVis can recognize the “Continent” attribute as a categorical variable and the “Population” attribute as a numeric variable automatically. From there, ShotVis recommends visualizing the data as a bar chart or line chart as shown at the bottom of Figure 9(c-d). However, since the names of the continents are pre-defined inside ShotVis, our system also recommends the map-based visualization.

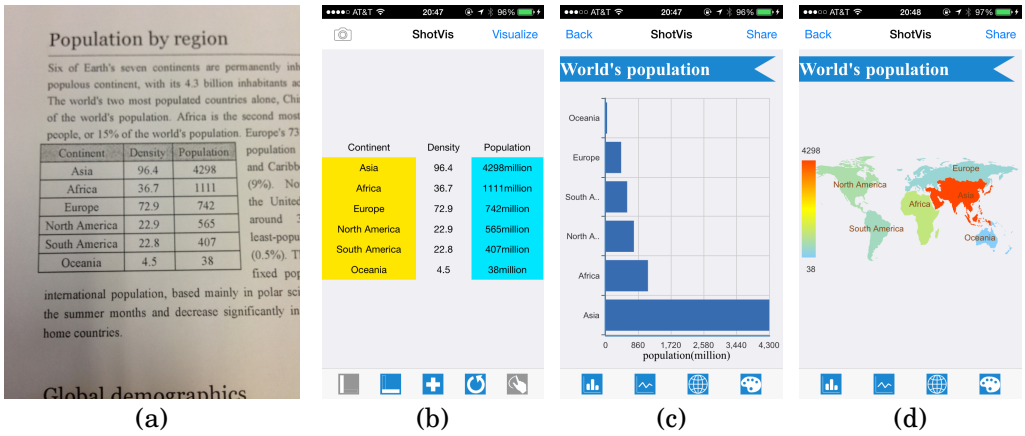


Fig. 9. (a) a photo with a tabulation of the world’s population. (b) after data manipulation. (c) bar chart visualization of the data. (d) map visualization of the data.

5. USER STUDY

ShotVis is designed to be used in mobile environments rather than on the desktop. We have conducted a user study to evaluate both the usability and effectiveness of ShotVis. Each participant was given 3 distinct pieces of printed textual data and asked to create two visualizations for each piece of data with ShotVis and with visualization tools they are familiar with on the PC. After completing the tasks, a questionnaire was given to evaluate the user experience of ShotVis. Subjective feedback is also collected from the participants through an interview at the end of the experiment.

5.1. TASK AND QUESTIONNAIRE DESIGN

In the evaluation, 4 participants (2 male and 2 female) participated in this user study, with ages ranging from 25 to 27. All participants are skilled in using smartphones and familiar with touch interactions. The majors of the participants include Computer Science, Software Engineering and Aerospace Engineering.

No subject had previous experience with ShotVis. A short training was provided to teach them how to use ShotVis. The training consisted of a set of demo videos of the datasets used in Section 4 together with a real-time explanation from the instructor. After that, the participants were permitted to play with the datasets first and to ask the instructor any questions in order to make sure that they all fully understand ShotVis and were familiar with its interactions.

The task for the participants was to visualize the 3 printed documents used in Section 4. First, they were asked to visualize the data with any tool they are familiar with on a PC and then with ShotVis on a mobile Device. All the datasets were provided as printed documents, except the ”paper” one. For the ”paper” dataset, they were free to choose either the printed or digital edition.

With ShotVis, participants were asked to examine whether there were mistakes in the recognized data and fix the mistakes (if any were found). After that, they can freely manipulate and visualize the data. With the PC, there was no constraint and they were allowed to use any visualization tool they were familiar with. However, for tasks that required using printed documents as input, participants were asked to transfer the data from the printed documents to digital ones using either OCR software or by hand.

After finishing the tasks, the participants were asked to answer a questionnaire with 5 questions (Q1-Q5) on a 5-point scale (1=very positive, 5=very negative). The questions are listed below:

Q1. Is it easy to learn how to use ShotVis?
Q2. Are the interactions in ShotVis intuitive?
Q3. Is it more efficient to make visualizations with ShotVis than with a PC?
Q4. Is it easy to make a visualization you are satisfied with in ShotVis?
Q5. Will ShotVis be useful in daily life and for common users?

5.2. RESULTS

Completion time for the visualization of each case was collected for evaluation for both the ShotVis and PC tools. We also recorded the time the participants spent entering the data into their computer. As we can see in Figure 10(a), for the first two tasks, the time spent in ShotVis is much shorter than that spent in entering the printed documents into PC, let alone the whole time spent with PC. While the result is not surprising as ShotVis was designed for this task, it further demonstrates the need for such a tool in the case where digital data is not readily available. For the last case, the “paper” one, no participant choose to input the data by hand, all of them used the

digital data, but the time spent in visualizing this data is still much longer with the PC than with ShotVis.

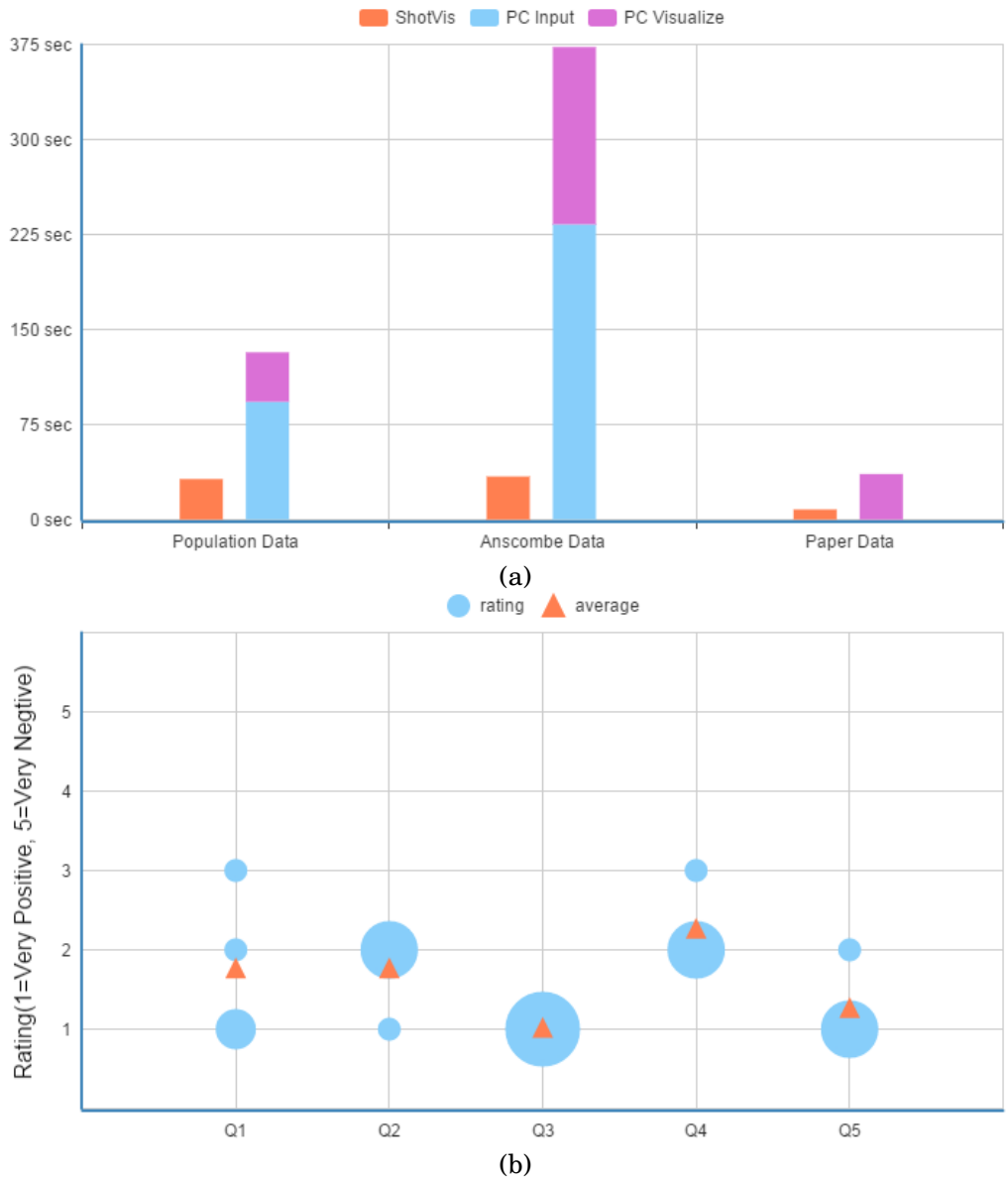


Fig. 10. (a) Time spent in each visualization task. The orange bar shows the average time spent when visualizing with ShotVis. The blue and purple bars show the average time spent entering data and making visualizations with ShotVis and the PC respectively. (b)The rating of the 5 questions in the questionnaire. The number of responses is encoded in the radius of the circle. The orange triangles represent the average rating for each question. We chose this representation of the results due to the low number of subjects in our study.

Besides the time, we also collected the visualizations that participants made both with a PC and with ShotVis. In Figure 11, each column corresponds to one case, the last row is the visualization made with ShotVis while other rows are those made with a PC. We have labeled the time cost of each visualization. From the results, we can see that compared to the conventional way we make visualizations (i.e. with PC), ShotVis can usually achieve equal or better results faster.

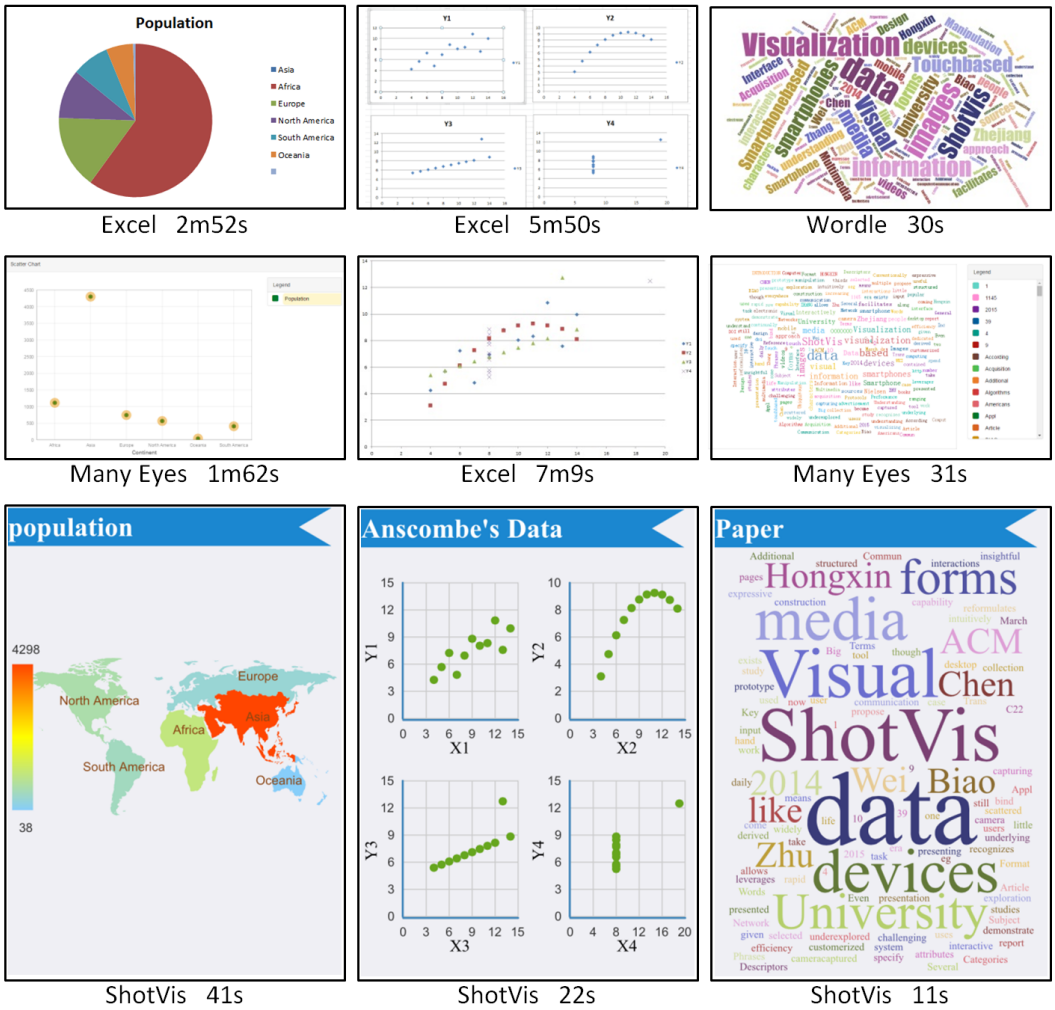


Fig. 11. Visualizations made by participants. Each column corresponding to one case. First two rows are made with PC, last row is made by ShotVis. Each visualization is labeled with the tool by which it is made and the time cost to make it.

The result of the questionnaire is shown in Figure 10(b). As we can see, the average ratings are all below 3. For the first two questions, the average ratings are both below 2, demonstrating that ShotVis is intuitive and easy to learn. For the third question about the efficiency of ShotVis compared to PC, all participants thought that it is quite efficient to complete the visualization with ShotVis. Good efficiency implies a strong usability of ShotVis. The rating for question 5 also implies that ShotVis was easy to

use as most participants think it will be quite helpful in their daily life. For question 4, it was the highest rating, indicating relatively low satisfaction with the visualizations created. This implies that more work can be done in the visualization template design.

5.3. LIMITATIONS AND LESSONS LEARNED

ShotVis bridges the data acquisition gap present in print media to enable visualization in everyday situations. However, ShotVis is still in the prototype phase and has several limitations. ShotVis can handle tabular data and purely text data(for the word cloud example), but for more unstructured data, like numbers from different parts of an article, it will fail. Even in the case of tabular data, when the table cannot be captured by a single image, it will be difficult for users to manipulate the data into an appropriate digital form. In order to make ShotVis more usable, we need to introduce more efficient interactions for smart-phones, thus making it easier to handle unstructured and large data. It will also be helpful to leverage advanced techniques in the field of machine learning to reduce OCR errors, especially when capturing hand-written tables from whiteboards.

6. CONCLUSIONS AND FUTURE WORK

ShotVis, is a touch-based data manipulation and visualization design system focusing on smartphones. ShotVis facilitates the construction of expressive and customized visualization from data derived from camera-captured images. As smartphones are becoming all-in-one mobile devices possessed by common users, our approach integrates data accquisition and visualization together that leverages the capabilities of smartphones with the representation efficiency of visualization.

ShotVis provides a short pipeline from unorganized image data to well-organized visual forms. Moreover, it allows users to interactively bind visual forms to the underlying data and specify visual attributes of selected forms with touch-based interactions. ShotVis facilitates understanding and interactive exploration of print media using camera-captured images via smartphone devices. Several case studies have been presented to demonstrate the effectiveness of ShotVis. According to our user study, most users provided positive feedback about ShotVis and they were eager to use this new solution.

Currently our prototype is just a first step towards the ideal model of mobile visual computing. Our approach is still limited by the recognition error rate and response speed of OCR. In the future, we will explore the possibility of enhancing the OCR portion and providing more content-aware factors. We also think it could be very interesting to integrate our approach with augmented reality applications such as with google glasses and Facebook oculus. In conclusion, we believe that the way to author visual media is changing, not only because of the nature of smartphones, but also due to the everyday needs of users.

REFERENCES

Stefano Burigat, Luca Chittaro, and Silvia Gabrielli. 2006. Visualizing locations of off-screen objects on mobile devices: a comparative evaluation of three approaches. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. ACM, 239–246.

Fabio Buttussi and Luca Chittaro. 2008. MOPET: A context-aware and user-adaptive wearable system for fitness training. *Artificial Intelligence in Medicine* 42, 2 (2008), 153–163.

Luca Chittaro. 2006. Visualizing information on mobile devices. *Computer* 39, 3 (2006), 40–45.

Bernd Girod, Vijay Chandrasekhar, David M Chen, Ngai-Man Cheung, Radek Grzeszczuk, Yuriy Reznik, Gabriel Takacs, Sam S Tsai, and Ramakrishna Vedantham. 2011. Mobile visual search. *Signal Processing Magazine, IEEE* 28, 4 (2011), 61–76.

Jie Hao and Kang Zhang. 2007. A mobile interface for hierarchical information visualization and navigation. In *Consumer Electronics, 2007. ISCE 2007. IEEE International Symposium on*. IEEE, 1–7.

Mark Harrower and Cynthia A Brewer. 2003. Colorbrewer. org: an online tool for selecting colour schemes for maps. *The Cartographic Journal* 40, 1 (2003), 27–37.

Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2011. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3363–3372.

Eiman Kanjo, Steve Benford, Mark Paxton, Alan Chamberlain, Danae Stanton Fraser, Dawn Woodgate, David Crellin, and Adrain Woolard. 2008. MobGeoSen: facilitating personal geosensor data collection and visualization using mobile phones. *Personal and Ubiquitous Computing* 12, 8 (2008), 599–607.

SungYe Kim, Ross Maciejewski, Karl Ostmo, Edward J. Delp, Timothy F. Collins, and David S. Ebert. 2008. Mobile Analytics for Emergency Response and Training. *The Journal of Information Visualization* 7, 1 (2008), 77–88.

Fabrizio Lamberti and Andrea Sanna. 2007. A streaming-based solution for remote visualization of 3D graphics on mobile devices. *Visualization and Computer Graphics, IEEE Transactions on* 13, 2 (2007), 247–260.

Hyunjoon Lee, Eli Shechtman, Jue Wang, and Seungyong Lee. 2014. Automatic Upright Adjustment of Photographs With Robust Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 5 (2014), 833–844.

Richard G. Lyons. 2010. *Understanding Digital Signal Processing (3rd Edition)*. Prentice Hall.

Elsa Macías, Hanna Abdelfatah, Alvaro Suárez, and Alejandro Cánovas. 2011. Full geo-localized mobile video in Android mobile telephones. *Network Protocols and Algorithms* 3, 1 (2011), 64–81.

Elsa MaciasMacías, Jaime Lloret, Alvaro Suarez, and Miguel Garcia. 2012. Architecture and protocol of a semantic system designed for video tagging with sensor data in mobile devices. *Sensors* 12, 2 (2012), 2062–2087.

Microsoft. 2014. Office Lens: A OneNote scanner for your pocket. (March 2014). http://blogs.office.com/2014/03/17/office-lens-a-onenote-scanner-for-your-pocket/

Gary Miner, John Elder, Andrew Fast, Thomas Hill, Robert Nisbet, and Dursun Delen. 2012. *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. Academic Press.

Tamara Munzner. 2014. *Visualization Analysis and Design*. A K Peters/CRC Press.

Antti Oulasvirta, Tye Rattenbury, Lingyi Ma, and Eeva Raita. 2012. Habits make smartphone use more pervasive. *Personal and Ubiquitous Computing* (2012), 105–114.

A. Razip, A. Malik, S. Afzal, S. Joshi, R. Maciejewski, Y. Jang, N. Elmqvist, and D. S. Ebert. 2014. A Mobile Visual Analytics Approach for Situational Awareness and Risk Assessment. In *Proceedings of IEEE PacificVis*.

Jonathan C. Roberts, Panagiotis D. Ritsos, Sriram Karthik Badam, Dominique Brodbeck, Jessie Kennedy, and Niklas Elmqvist. 2014. Visualization beyond the Desktop-the Next Big Thing. *IEEE Computer Graphics and Applications* 34, 6 (2014), 26–34.

Manolis Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and Jeffrey Heer. 2011. Revision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 393–402.

Rosália G. Schneider and Tinne Tuytelaars. 2014. Sketch Classification and Classification-driven Analysis Using Fisher Vectors. *ACM Trans. Graph.* 33, 6, Article 174 (2014), 9 pages.

Richard Szeliski. 2010. *Computer Vision: Algorithms and Applications*. Springer.

Andrey Vasilev, Ilya Paramonov, Sergey Balandin, Ekaterina Dashkova, and Yevgeni Koucheryavy. 2012. Context capturing in smart space applications. *Network Protocols and Algorithms* 4, 4 (2012), 84–100.

Yao Wang, Jeorn Ostermann, and Ya-Qin Zhang. 2001. *Video Processing and Communications*. Prentice Hall.

Mark Weiser. 1999. The Computer for the 21st Century. *SIGMOBILE Mob. Comput. Commun. Rev.* 3, 3 (July 1999), 3–11.

Wikipedia. 2014. Optical character recognition. (Nov. 2014). http://en.wikipedia.org/wiki/Optical_character_recognition

J.O. Wobbrock, A.D. Wilson, and Y. Li. 2007. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '07)*. ACM, 159–168.

Hee Yong Yoo and Suh Hyun Cheon. 2006. Visualization by information type on mobile device. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation-Volume 60*. Australian Computer Society, Inc., 143–146.

Hong Zhou, Huamin Qu, Yingcai Wu, and Ming-Yuen Chan. 2006. Volume visualization on mobile devices. In *14th Pacific conference on computer graphics and applications*. 76–84.

Received December 2014; revised March 2015; accepted June 2015