VISUAL 2017

The Second International Conference on Applications and Systems of Visual Paradigms

ISBN: 978-1-61208-577-7

July 23 - 27, 2017

Nice, France

VISUAL 2017 Editors

Pascal Lorenz, University of Haute Alsace, France

Lucio De Paolis, University di Salento, Italy

VISUAL 2017

Foreword

The Second International Conference on Applications and Systems of Visual Paradigms (VISUAL 2017), held between July 23 - 27, 2017 - Nice, France continued the inaugural event in putting together complementary domains were visual approaches are considered in a synergetic view.

Visual paradigms were developed on the basis of understanding the brain's and eye's functions. They spread over computation, environment representation, autonomous devices, data presentation, and software/hardware approaches. The advent of Big Data, high speed images/camera, complexity and ubiquity of applications and services raises several requests on integrating visual-based solutions in cross-domain applications.

We take here the opportunity to warmly thank all the members of the VISUAL 2017 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to VISUAL 2017. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the VISUAL 2017 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that VISUAL 2017 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the area of visual oriented technologies.

We are convinced that the participants found the event useful and communications very open. We also hope that Nice provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city

VISUAL 2017 Chairs:

VISUAL Steering Committee

Vijayan K. Asari, University of Dayton, USA
Robert S. Laramee, Swansea University, UK
Luciano Pereira Soares, Insper, Brazil
Jibonananda Sanyal, Oak Ridge National Laboratory | Mississippi State University, USA
Irene Yu-Hua Gu, Chalmers University of Technology, Sweden

VISUAL Industry/Research Advisory Committee

Kresimir Matkovic, VRVis Research Center, Vienna, Austria
Jinrong Xie, eBay Inc., USA
Afzal Godil, National Institute of Standards and Technology, Gaithersburg, USA
Silvia Biasotti, Consiglio Nazionale delle Ricerche, Genova, Italy
Carolin Helbig, Helmholtz Centre for Environmental Research (UFZ), Germany

VISUAL 2017

Committee

VISUAL Steering Committee

Vijayan K. Asari, University of Dayton, USA
Robert S. Laramee, Swansea University, UK
Luciano Pereira Soares, Insper, Brazil
Jibonananda Sanyal, Oak Ridge National Laboratory | Mississippi State University, USA
Irene Yu-Hua Gu, Chalmers University of Technology, Sweden

VISUAL Industry/Research Advisory Committee

Kresimir Matkovic, VRVis Research Center, Vienna, Austria
Jinrong Xie, eBay Inc., USA
Afzal Godil, National Institute of Standards and Technology, Gaithersburg, USA
Silvia Biasotti, Consiglio Nazionale delle Ricerche, Genova, Italy
Carolin Helbig, Helmholtz Centre for Environmental Research (UFZ), Germany

VISUAL 2017 Technical Program Committee

Athos Agapiou, Eratosthenes Research Centre / Cyprus University of Technology, Cyprus
Konstantin Aksyonov, Ural Federal University, Russia
Djamila Aouada, University of Luxembourg, Luxembourg
Vijayan K. Asari, University of Dayton, USA
George Baciu, The Hong Kong Polytechnic University, Hong Kong
Yufang Bao, Fayetteville State University, USA
Yannick Benezeth, Univ. Bourgogne Franche-Comté, France
Stefano Berretti, University of Florence, Italy
Silvia Biasotti, Consiglio Nazionale delle Ricerche, Genova, Italy
Hans-Peter Bischof, Rochester Institute of Technology, USA
Kadi Bouatouch, IRISA / Unviversity of REnnes 1, France
Kevin Chalmers, Edinburgh Napier University, UK
Bin Chen, Purdue University Northwest, Hammond, USA
Haeyong Chung, University of Alabama in Huntsville, USA
Anastasios Doulamis, National Technical University of Athens, Greece
Pierre Drap, Aix-Marseille University, France
Soumya Dutta, Ohio State University, USA
Zlatko Franjcic, Chalmers University of Technology, Gothenburg, Sweden
Daniela Giorgi, Institute of Infomation Science and Technologies (ISTI) - National Research Council of Italy (CNR), Pisa, Italy
Afzal Godil, National Institute of Standards and Technology, Gaithersburg, USA
Prashant Goswami, Blekinge Institute of Technology, Karlskrona, Sweden
Valerie Gouet-Brunet, IGN, France

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Visual Deep Learning Recommender System for Personal Computer Users

Daniel Shapiro*†, Hamza Qassoud*†, Mathieu Lemay† and Miodrag Bolic*
*School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Ontario, Canada
Email: {dshap092, hqass076, mbolic}@eecs.uottawa.ca
†Clockrr Inc., and Lemay Solutions Consulting Inc., Ottawa, Ontario, Canada
Email: {daniel, matt}@lemaysolutions.com

***Abstract*—This work presents a new architecture for creating virtual assistants on personal computers, building upon prior work on deep learning neural networks, image processing, mixed-initiative systems, and recommender systems. Recent progress in virtual assistants enables them to converse with users and interpret what the user sees. These systems can understand the world in intuitive ways with neural networks, and make action recommendations to the user. The assistant architecture in this work is described at the component level. It interprets a computer screen image in order to produce action recommendations to assist the user. It can assist in automating various tasks such as genetics research, computer programming, engaging with social media, and legal research. The action recommendations are personalized to the user, and are produced without integration of the assistant into each individual application executing on the computer. Recommendations can be accepted with a single mouse click by the computer user.**

Keywords–Recommender systems; Image processing; Deep learning.

I. INTRODUCTION

Personal computers typically push the cognitive effort of solving a problem or discovering a task onto the user, requiring the user to take the initiative and tell the computer how to solve the problem based upon hints provided by the computer. For example, a search engine box presents a blank slate to the user, who must decide what to type into the box, and similarly, an error message can lead the user to search in a search engine for a solution, rather than the computer offering a solution to the problem.

Virtual assistants can take many forms, including assistance hardware appliances [1], cloud-based assistants [2], and application specific assistants [3]. Advances in artificial intelligence have enabled the user to express her intent using voice commands [4]–[12]. Additional recent advances in mobile computing such as Google Now on Tap enable a mobile phone to interpret a picture of the phone's screen and recommend relevant actions and information [13] [14]. This type of image processing recommender system represents a step in the right direction towards intelligent anticipatory computing. However, Google Now on Tap is not available for personal computers, where many users work, and it is not a proactive system. In other words, Google Now on Tap waits for the user to request recommendations by pressing a button. With the exception of reminders, personal computer assistants, such as [15] and [16], were designed to require the user to express intent in order to access a recommendation [17].

Anticipatory computing with artificial intelligence can assist users in real-time as they interact with personal computers [18]. However, the basic mismatch in initiative between the computer and user remains unresolved. Virtual assistants for personal computers can converse with users more naturally than ever, but cannot see what the user sees, or take the initiative to understand the computer screen in some intuitive way and provide action recommendations to the user. Current efforts revolve around extracting the intent of the user from the user's input, transforming the identified intent into a query, and then returning query results to the user. Some systems go further and attempt to support interactive dialog, reminders, and application launching. This command and dialog approach to virtual assistants continues to place the onus of specifying intent squarely onto the user. It is that cognitive pressure applied to user by the assistant - the need to have the user specify what the user wants - that is a limitation of existing works.

To reduce the cognitive pressure on the user, this work presents a virtual assistant called Automated Virtual Recommendation Agent (AVRA). AVRA follows a Mixed-Initiative (MI) approach to human-computer interaction, where a human and virtual agent work together to achieve common goals. The approach is to offload to AVRA some of the cognitive pressure of understanding onscreen problems and goals visible on the computer screen, and recommending actions to the user as solutions.

At the heart of AVRA is a Recommender System (RS) which provides action recommendations to the user on its own initiative. This approach is different from contemporary user assistance software, which typically provides a text box for the user to enter a command as in [19], voice recognition to speak commands to the computer as in [20], or other such user-driven computer interaction mechanisms. The difference in AVRA is that there is no dialog for user input. AVRA observes the visible information on the computer screen to produce recommendations on its own, and the user can accept or reject the advice from AVRA.

The contribution of this work is to describe the architecture of a virtual assistant with shallow application integration. Related work is discussed in Section III. The process of training AVRA to recognize contexts and keywords is described in Section IV. In Section V, AVRA is evaluated using execution time, recall, precision, precision-recall and Receiver Operating Characteristic (ROC) curves as metrics. Section VI contains a summary of this work and a discussion of future research directions.

II. SYSTEM OVERVIEW

AVRA follows a series of steps in order to produce recommendations. The primary goal of providing action recommendations based upon an image of the computer screen leads to several sub-tasks such as context identification (Section II-A),

 ISBN: 978-1-61208-577-7

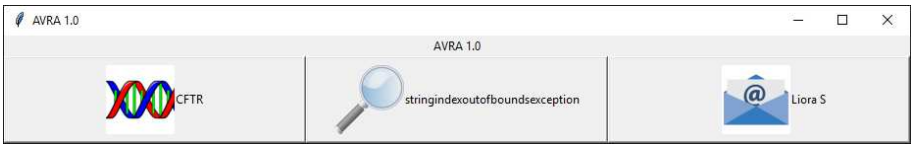

Figure 1. AVRA's graphical user interface showing recommendations to explore a specific gene name, a Java programming error, and compose an e-mail.

context-specific text filtering (Section II-B), context-specific character-level classification of onscreeen text (Section II-C), and objective ranking of all possible recommendations and re-ranking these recommendations according to the user history (Section II-D).

To provide the user with action recommendations, AVRA begins by detecting the particular graphical contexts appearing on the computer screen. This is followed by the detection of tasks within those contexts for which the RS knows how to help the user. Next, it ranks the recommendations for the user so that they are customized to the user's past behavior. Finally, the RS presents three action recommendations to the user via a Graphical User Interface (GUI), as shown in Figure 1. The buttons are decorated with meaningful icons, making it clear what action recommendations are being offered to the user, and reducing the footprint of the GUI on the screen. Further insight into why the recommendation was made is available as a tooltip when the user's mouse scrolls over a GUI button. Interaction with the GUI is very simple, limiting the scope of the interaction between the RS and the user. When the user clicks on one of the GUI buttons, the action corresponding to the solution recommendation in that button is executed. These actions may be email composition, opening a browser window to a particular website, opening a document, etc.

Action recommendations are presented as binary choices in the GUI, limiting the scope of user feedback to clicking or not clicking buttons. This simple interaction makes the user experience significantly more intuitive than a voice command system, because the user does not know with certainty when issuing a voice command that the computer will comply correctly with the command, whereas AVRA advertises what actions it proposes to perform prior to the user commanding the action to be performed. Furthermore, the GUI only updates the recommendations when the mouse is stationary and not hovering over the GUI. This way, if the user sees a recommendation in the GUI and begins to move the mouse toward the buttons, the recommendation will not be replaced, as this would likely annoy the user. Similarly, if the user is pondering a recommendation (perhaps reading the tooltip text while hovering the mouse over a button in the GUI), the action recommendation should not be replaced. This type of confidence building is important in MI system design and is discussed in [21] and [22].

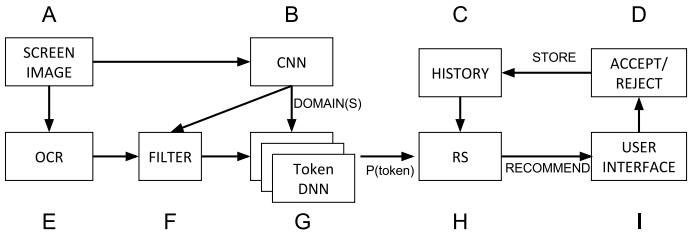

Figure 2. AVRA System Overview.

Like many artificial intelligence platforms, AVRA s implemented as a web service. Each client repeatedly captures an image of the screen which is then interpreted by a server-side program. The server processes the image by parsing and classifying the text and graphics on the computer screen. The server responds to the client's image submission with a set of 3 personalized action recommendations presented to the user.

The client consists of components A and I in Figure 2, while the server implements components $B - H$. The advantages of placing most system components on the server in a thin-client design include the ability to leverage server-side GPU acceleration, having the user profile accessible from multiple computers as a user moves from machine to machine, and having access to the user profiles of many users for future work on collaborative filtering of recommendations.

This approach to assisting the user generalizes to non-programming domains where the user researches items that appear onscreen. For example, when the user is browsing social media, the RS can detect the names of friends and recommend an action such as composing an email to the detected contact when that friend's name appears onscreen. Another example developed in AVRA is genetic research, where the user is reading a PDF document involving genetic research. The RS can detect this genetic research context, and propose to open a browser window to a web page detailing the relationship between a gene name recognized on the computer screen and other genes.

A. Context Identification

A computer screen image may contain graphical information regarding several topics. For example, a terminal window alongside a browser window conveys graphical information on these topic. Each topic detected in the computer screen image is called a context, and each context is recognized by a trained Convolutional Neural Network (CNN). The image features within a context can be the contents of a program such as a document format, or the look and feel of a program such as menus and color schemes. Figure 2 shows that each screen capture is parsed by a CNN (Figure 2 B) to identify what contexts should be explored for each image. The context recognition CNN that steers recommendation generation is not tightly integrated into each specific application producing onscreen information. Rather, an image of the computer screen is processed by the CNN classifier to detect contexts, and these detected contexts guide text filtering and classification units, which search the onscreen text for keywords based upon which AVRA can recommend actions. This shallow integration of an RS into the programs executing on a personal computer system is novel. Because of this shallow integration approach, AVRA has visibility over all programs visible to the user, and can therefore help with multitasking, whereas an isolated program restricted to text or voice commands has insufficient visibility into the workspace (i.e., the computer screen) to make such recommendations. Effectively, AVRA sees what the user sees.

In AVRA, the classifier can make more than one prediction per image. Multiple different contexts can be present on the

computer screen at the same time, and therefore consider that one image of the computer screen containing side-by-side windows of the Eclipse IDE and a console window could trigger two predictions which may both be accurate.

Through supervised learning, AVRA is initialized with knowledge of context-detection for certain domains, context-specific textual terms (called keywords), and action recommendations corresponding to each possible context-specific keyword detection.

B. Text Filtering

Optical Character Recognition (OCR) (Figure 2 E) converts the screen image (Figure 2 A) into a string of text to be parsed. The raw text from the OCR system (Figure 2 E) is cut into many small text segments, which are then very quickly filtered (Figure 2 F), selecting for a minimum similarity between the candidate text and keywords of each detected context (Figure 2 B). This filtering step is necessary to reduce the number of text segments reaching the relatively slow text classifier (Figure 2 G).

C. Character-level Text Classification

A Deep Neural Network (DNN) for each activated context (Figure 2 G) processes the text output of the text filtering stage (Figure 2 F) using character-level neural network classification to detect keywords within the text that are associated with recommendations stored in the RS database. Each DNN is trained by processing an image of the keyword text through OCR software (generated using a text-to-image generator), and then encoding the results into vectors that include letter frequency information. The training process is described in [23], and the goal of the approach is to robustly detect keywords even in the presence of OCR output spelling errors and frame-shifts in the text.

D. Recommender System

The recommendations generated by the DNNs are filtered (Figure 2 H) to identify the top 3 ranked recommendations to be presented in the user interface (Figure 2 I). The user may accept recommendations by clicking a button in the user interface, causing the corresponding action(s) to be executed on the computer (Figure 2 D) and the user history is also updated (Figure 2 C).

AVRA provides personalized recommendations that adapt over time to the changing behavior of the user. The RS learns from the history of user interactions with AVRA to adapt recommendations to the user's needs. Employing a virtual assistant in this way is a novel approach to helping the user with tasks. In the programming example, AVRA can share with the user the task of identifying the error message on the computer screen, looking up actions which may help the user with the error, and identifying the most relevant action recommendations. This approach puts little or no pressure on the mind of the user in the course of normal computer use.

AVRA includes a probabilistic hybrid RS model, where the RS model combines the classification confidence level associated with each action recommendation with the confidence level associated to the each context detected on the computer screen. These two confidence levels are further combined with the user's preferences (history of clicks and rejections of action recommendations). Following this adaptive RS approach, the recommendations generated by a given image of the computer screen will change over time based upon the actions of the user.

III. Related Work

Using reinforcement learning, specifically a deep Q-network, [24] developed a deep learning artificial intelligence system that can play many different video games using computer screen images as input and joystick positions as the output. The breadth of different applications played by the same neural network was an impressive validation of the power of reinforcement learning. The overlap between AVRA and [24] is the use of a CNN to process the image of the screen, and then using fully connected layers of a DNN to make a decision. In the case of AVRA, the DNN output is a recommendation to be ranked, whereas in [24] the DNN outputs represent joystick positions. Also, in [24] the goal of the CNN in the system was regression, and the CNN did not perform pooling (down-sampling), while in contrast AVRA's CNN does perform down-sampling via pooling, as the AVRA CNN is outputting a classification rather than a target score. Furthermore, unlike [24], AVRA's CNN is connected to multiple DNNs in a low resolution way (either activating a DNN or not), while the CNN from [24] were connected directly into the fully connected layers of the DNN.

The shallow integration between AVRA and the applications executing on the user's computer relates to prior work on high-level query processing proposed in [25]. The concept in [25] is that shallow clues in a query can hint at the correct databases to search. Therefore, a query to many databases need not be written for each database; rather, a high-level intermediate query engine can dynamically steer the query to the right databases. In AVRA, the shallow search is performed by a CNN to detect contexts, and then DNNs, along with filtering algorithms and OCR, perform deeper context-driven analysis. The key information to understand from [25] in this work is that the low-level context by context search was bypassed, and instead a dynamic high-level context search was achieved.

A. Recommender Systems

An RS must score and rank recommendations in order to present high-quality options to the user. Content-Based Filtering (CBF) is an approach where the recommendation score is increased if related items were rated positively in the past. CBF makes it more likely for items and topics preferred in the past to be recommended in the future [26] [27]. As in [28], this work takes the approach to modify the score of a recommendation based on both topic and item similarity. Similar to [29], in which the user's tagging history informs the likelihood of recommending an item, AVRA uses the user's history of accepting and rejecting recommendations to modify the likelihood of new recommendations over time. Specifically, action recommendations in AVRA are biased toward actions and contexts where the user clicked to accept a recommendation. To a smaller extent action recommendations are negatively biased toward recommendations that were not accepted by the user. AVRA includes a probabilistic hybrid RS model similar to [30].

Combining CBF with a taxonomic record of user preferences improves the quality of recommendations [31]. AVRA

 ISBN: 978-1-61208-577-7

follows this approach by scoring recommendations based upon context, while taking into account user preferences. Similarly, [32] combines CBF with knowledge of the domain. Ontology and taxonomy approaches to modeling recommendation scores encapsulates entity relationships. An ontology-based approach could be incorporated into AVRA's RS in addition to the described context-based CBF approach.

Context-aware recommender systems can focus on using the context of the user (as in [33] in which the user profile can guide the recommendation score) and/or the context of the recommendation (as in [34]) to modulate recommendation scores. In this work, context is approached from the perspective of [34], where recommendations can be annotated with additional situation-based information called the context. Rather than interpreting the computer screen in every possible context, this work is about narrowing down the search for onscreen meaning to a select few contexts.

The cold start problem is a situation where the RS has insufficient information about a user to make high-quality recommendations [35]. This quality problem is less pronounced in AVRA as the recommendations are strongly driven by onscreen activity, which leads naturally to an initial set of recommendations that are adjusted over time based upon clicks or recommendation rejections. The review article [35] provides an overview of the RS state of the art. RS recommendation quality is discussed in the literature, and some key ideas are repeated here. Reverb [36] is an IDE plug-in that recommends previously visited web pages related to code being written by a programmer in the IDE. The idea is to reduce false-positive recommendations when offering the programmer a recipe to look up as she is writing a program. This increase in quality is accomplished by only recommending pages that have been previously visited by the user. A predecessor of Reverb is Fishtail [37], which recommends web pages for the same purpose, but without restricting recommendations using the user history as Reverb does. The consequence of recommending pages based only upon keywords is lower quality recommendations [38]. Another interesting example from application-specific RS is [3], an MI system that works in tandem with the programmer to expose programming recipes and other useful programming tips. Whereas [3] [37] [36] and others integrate into specific programs, AVRA was designed to process images of the computer screen as a whole, loosening the integration between the applications and RS.

An RS for personal computers that learns and predicts actions was described in [39]. In that work, a distinction is made between types of information extracted by the RS: action features describe items that happened recently on the computer, while state features describe the current state of the machine. Examples of action features are a history of program calls, the stream of keyboard characters, and onscreen streaming video. Examples of state features are a list of programs that are currently running, the current directory, and the current language settings. In [40], the goal of the RS is prediction of the next command to be typed into a terminal program. Recommendations from this type of RS are based upon previous input (action features), but cannot see the current or past output resulting from the execution of these commands. AVRA seeks out information from many different programs visible onscreen, while command prediction only detects the command history within one program. An unrelated feature in [40] that influenced the design of AVRA is the mapping of recommendations to function keys. In [40] the keyboard keys F1 to F5 were mapped to five different command recommendations presented onscreen to the user. Picking up on this interface design, AVRA uses three onscreen buttons to expose recommendations.

The RS user interface design and behavior is a major driver of user adoption, and is a factor in the user's trust of the RS recommendations [38]. Furthermore, exposing the capability to explore and comprehend recommendations in the GUI can be a negative influence on the user experience [38]. The virtual assistant "Clippy" from Microsoft Word is an example of good RS design combined with bad GUI design [38]. Clippy was considered by many users to be too intrusive, even though the recommendations provided were typically useful [38]. In contrast, less intrusive features such as spellcheck, autocomplete and autocorrect are now widely adopted in many technologies. Two independent factors influencing GUI design in RS are obviousness of the recommendations and cognitive effort required to use the interface [38]. Cognitive effort in AVRA is reduced by limiting the number of buttons in the GUI, restricting the action recommendation type, and restricting the complexity of the action recommendations. The obviousness of the recommendations in the GUI is enhanced by the graphical icons used to symbolize various tasks. For example, a mail icon next to a person's name (e.g., "Daniel") is more obvious than the sentence "Compose an e-mail to Daniel". The former contains only one word (the name), and can be interpreted at a glance without reading into the text of the recommendation with full focus.

AVRA was designed to continuously prompt the user. Similarly, [41] presents a smart home application for assisting the elderly that involves prompting the user many times. To limit the negative effects of repeated prompting, the system learns rules that define when activities normally occur and utilizes these rules to automate prompting. Whereas [41] focuses on waiting to prompt until the notification is needed, AVRA focuses on recommending quickly based upon the current content on the computer screen, or not at all.

B. Mixed Initiative

Iterative sensemaking is a process of working with data sets of semi-reliable structure to produce intermediate results along the way to a conclusion [42] [43]. Each sensemaking iteration involves two steps: foraging and synthesis. These steps are easier to perform quickly with assistance from a predictive intelligent system, such as an MI system. An MI system assists a human analyst to derive and take advantage of insights into data. In MI systems, the breakdown of work between the computer and the human focuses on the strength of each participant in the iterative problem solving activity [44]. The goal of the collaboration between human and virtual agent is producing insights leading to one or more conclusions regarding the data [22]. AVRA's design was based upon recent work that recommends the desired characteristics for MI systems [45] [22].

Graphics-based interaction with the MI system helps the user to more effectively concentrate on decision making [46]. AVRA's 3 button UI design was influenced by similar interface designs, including the "Smart Reply" feature of Google Inbox, which offers up to 3 candidate email message replies as draft

 ISBN: 978-1-61208-577-7

responses [47]. Unlike Smart Reply, AVRA's UI needs to specify multiple items per button: an action, a short snippet of text describing the data used by the action, and a longer tooltip message about the recommended action. The representation of these 3 items per button is accomplished using action icons in the GUI rather than action description text, saving real estate in the GUI that is better utilized by the action data description text. The tooltip hides the longer full-text description of the recommended action. The tooltip text feature allows the user to glean additional insight by reading a longer text explanation for the proposed action recommendation. Having compacted the form factor of the GUI, there is a tendency to increase the number of GUI buttons. However, including too many buttons in the GUI would create a paradox of choice where the interface begins adding cognitive pressure onto the user by forcing her to think about many possible options, rather than achieving the goal this system was designed for, which is to reduce that cognitive pressure [48].

IV. Recommendation Learning

AVRA contains processes for supervised learning of new contexts through CNN training, and for learning new keywords for existing contexts through DNN training. The RS associates each DNN keyword with one or more recommendation actions. AVRA can also learn new information from unsupervised learning of contexts and keywords.

A. Supervised Context Learning

One expects that a CNN recognizing activities on the computer screen requires thousands of classes to represent the wide variety of activities taking place on the computer. Training the CNN through supervised learning to recognize a new context requires many example images. Typically, several hundred representative images of a context are required to train the CNN. To acquire images that look like a particular context (e.g., the Eclipse IDE) an Internet image scraper was used to pull many high-resolution images from online image search engines based upon DNN keywords within the context, and adjusted these keywords manually to improve the quality of the scraped images (e.g., "eclipse IDE java programming"). The scraper was implemented in nodejs using the images-scraper library [49], and filtered results by file type and file name to exclude unhelpful or damaged images. The resulting many thousands of images for each context were reduced with automated duplicate image deletion, followed by further manual inspection. Images were selected that best represent the context to the person selecting the images. A second source of training data was locally generated training images. In order to test out the feasibility of unsupervised learning, AVRA's image capture program was executed for several days in the background as the computer was used for normal work activities, in order to generate sufficient data to browse and extract relevant images. Once the data set of representative images was finalized, the supervised training was ready to proceed.

Cross-validation is the fitness measure used for supervised context learning during CNN training. Transfer learning was used to decrease CNN training time and increase classification accuracy [50]. The transfer learning was implemented using the inception v3 CNN [51], taking advantage of feature detection capabilities of image recognition software trained on large sets of images.

TABLE I. CNN CONFUSION MATRIX FOR K=1% RESULTED IN HIGH RECALL WITH LOW PRECISION.

		Predicted											Recall
		A	B	C	D	E	F	G	H	I	J	K	
Actual	A	5	5	5	0	4	4	0	0	0	0	0	1
	B	2	5	5	2	5	5	0	0	0	0	0	1
	C	0	3	5	0	5	5	0	0	0	0	0	1
	D	0	0	0	5	2	0	1	0	1	0	1	1
	E	0	4	1	2	5	0	0	0	0	0	0	1
	F	0	0	5	5	5	5	0	0	0	0	0	1
	G	0	0	0	0	0	0	5	0	0	0	4	1
	H	0	0	0	0	0	0	0	5	0	1	0	1
	I	0	0	0	0	0	0	3	1	5	4	0	1
	J	0	0	0	0	0	0	2	3	2	5	0	1
	K	0	0	0	0	0	0	3	0	0	0	5	1
Precision		.71	.29	.24	.36	.19	.26	.36	.56	.63	.50	.50	

To provide a motivating example of AVRA in action, the CNN was trained on 9654 images in total, covering 11 different contexts. A graph of the training and cross-validation accuracy at each epoch during the CNN training is presented in Figure 3, produced with inception's retraining code [51]. The retraining configuration included 4000 training steps. Figure 3 shows that validation accuracy lags training accuracy, as expected. The shape of the learning graph is also as expected, with an initial high rate of learning followed by slower incremental improvements in accuracy. The performance of the CNN was measured in terms of recall and precision when trained to identify several contexts. The testing images (5 images per class for 11 classes) were not included in the training dataset. The testing data had 1 class per image. The results are presented in a confusion table (Table I).

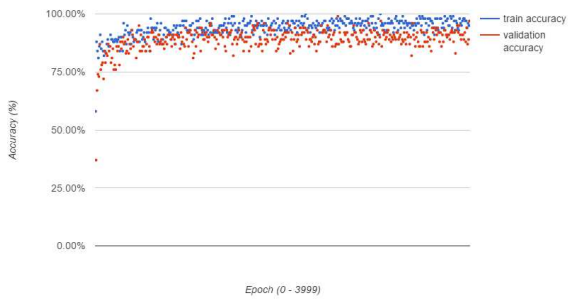

Figure 3. Training and validation classification accuracy during the 4000 epochs of the CNN training process. The final test accuracy was 92.20%

The testing dataset used to produce the confusion matrix of Table I only contained images with one class per image. Hyperparameter K of the CNN is a classification threshold that tunes AVRA to be more conservative (higher threshold) or more open to evaluating hypotheses (lower threshold). The recall observed in Table I is high at the expense of precision. There is a fundamental trade-off between recall and precision determined by the threshold K. This is observed in Table II, where K=95%. With such a high requirement for certainty that a context has been detected, there is a higher precision, but only when the context is detected. The cost of increasing K is missing the context completely, costing the algorithm on both the precision and the recall metrics. Because false positive context detection is generally less damaging to the recommendation quality than false negative context detection, AVRA is configured with a low K threshold as in Table I, rather than a high value as in Table II.

 ISBN: 978-1-61208-577-7

TABLE II. CNN CONFUSION MATRIX FOR K=95% RESULTED IN LOW RECALL WITH HIGH PRECISION.

	Predicted											Recall
		A	B	C	D	E	F	G	H	I	J	K	
Actual	A	0	0	0	0	0	0	0	0	0	0	0	0
	B	0	0	0	0	0	0	0	0	0	0	0	0
	C	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	3	0	0	0	0	0	0	0	0.6
	E	0	0	0	0	1	0	0	0	0	0	0	0.2
	F	0	0	0	0	0	0	0	0	0	0	0	0
	G	0	0	0	0	0	0	2	0	0	0	0	0.4
	H	0	0	0	0	0	0	0	5	0	0	0	1
	I	0	0	0	0	0	0	0	0	1	0	0	0.2
	J	0	0	0	0	0	0	0	0	0	2	0	0.4
	K	0	0	0	0	0	0	0	0	0	0	3	0.6
Precision		0	0	0	1	1	0	1	1	1	1	1	

B. Supervised Keyword Learning

The DNN for each context was trained to recognize keywords within filtered OCR output text. The purpose of the DNN is to detect keywords in the presence of OCR transcription errors. The classification task is accomplished through a process of learning the mistakes made by the OCR system. More specifically, AVRA generates test images for each keyword, feeding the images through the OCR software to obtain string data. Next, AVRA encodes the error-laden OCR string output as binary ASCII characters into numerical vectors, which include letter frequency information in the encoding scheme [23]. The input vectors for DNN training were produced using this approach, while the index of each keyword within the RS is the output vector corresponding to a particular recommendation.

C. Unsupervised Keyword and Context Learning

Generating recommendations from unsupervised learning involves learning CNN contexts as well as DNN keyword sets. AVRA's unsupervised learning function is to search for an onscreen keyword that the user searched for in the past (verbatim) after seeing it on the screen. This approach will be discussed in upcoming work.

V. Performance Evaluation

Execution time for the OpenMP and multicore approaches are reported here. Next, GP-GPU acceleration with CUDA is presented. Finally, recall, precision, precision-recall and ROC curves are presented.

1) OpenMP and Multicore DNN Training Acceleration: Each DNN in AVRA was trained for 150 epochs, and each epoch required approximately 250 seconds to complete on a low cost VM. This execution time is related to the size of the training and testing sets, and so these figures are different for each DNN. Assuming 50 DNNs with 150 epochs each, and an average epoch execution time of 250 seconds, the training time on a single VM would be: (50 DNNs) x (150 epochs) x (250 seconds) which is approximately 520 hours. This slow training and classification did not provide a reasonable response time at scale. Even if every DNN is trained on a different dedicated VM, the execution time becomes (150 epochs) x (250 seconds) which is approximately 10.5 hours. Reducing the number of epochs would reduce the accuracy of the DNN classification. This leaves the epoch execution time as the variable to be reduced.

OpenMP is a shared memory parallel programming API suitable for parallelizing applications on a multiprocessor computer [52]. Theano is compatible with OpenMP and ships with a benchmark for testing the speedup provided by OpenMP [53]. This benchmark was used to generate the data in Table III which is a generic view of accelerating theano with OpenMP.

TABLE III. THEANO'S OPENMP BENCHMARK, TIMED WITH A VECTOR OF 200,000 ELEMENTS AND 4GB RAM

OpenMP Threads	CPU Cores	Operation Type	Without OpenMP (s)	With OpenMP (s)	Speedup
1	2	Fast	0.000113	0.000099	1.14
2	2	Fast	0.000110	0.000066	1.67
3	2	Fast	0.000114	0.000137	0.83
4	2	Fast	0.000111	0.000128	0.87
1	2	Slow	0.006590	0.006091	1.08
2	2	Slow	0.006208	0.003060	2.03
3	2	Slow	0.006107	0.004127	1.48
4	2	Slow	0.006253	0.003738	1.67

TABLE IV. DNN TRAINING TIME PER EPOCH. THE AVERAGE BASELINE EXECUTION TIME WAS 257.

OpenMP Threads	CPU Cores	RAM (GB)	Epoch	Training Time (s)	Speedup from Average Baseline
1	2	4	0	263	N/A
1	2	4	1	251	N/A
2	2	4	0	196	1.3
2	2	4	1	220	1.2
3	2	4	0	151	1.7
3	2	4	1	173	1.5

The results from theano's benchmark indicate that configuring theano with 2 OpenMP threads is the best option for reducing epoch execution time. Perhaps this result is a product of the VM having 2 CPUs and OpenMP mapping one OpenMP thread to each CPU. However, testing with AVRA's DNN code reported in Table IV indicates that 3 OpenMP threads is best. This disagreement between the generic benchmark and the AVRA code itself is a reminder that often benchmarks are a poor proxy for testing performance improvements in computer architecture [54]. Note that even though a 2x speedup was achieved by using OpenMP, this only reduced the per-DNN execution time from 10.5 hours down to 5.25 hours ($> 150s$ / training epoch), which is still impractically slow. In the next section GP-GPU acceleration is leveraged to further reduce the per-epoch execution time of DNN training.

2) DNN Training Acceleration using GPU and CUDA: An AWS instance of type g2.2xlarge with a GRID K520 GPU was configured with an existing AMI containing theano for cuda 6.5 [55]. The system was tested with the same DNN code and data used to produce the data in Table IV, although OpenMP was not enabled.

CUDA is a parallel computing programming model which enables GPU hardware acceleration of computations [56]. The per-epoch execution time was reduced from 151 seconds without the using CUDA and the GPU, to 6.5 seconds when the GPU is used. This represents a 23x speedup relative to the best OpenMP result in Table IV. With a 6.5 second epoch execution time, training a DNN can take (6.5 seconds) x (150 epochs) which is approximately 16 minutes and 15 seconds. Note that the epoch execution time is highly variable based upon the size of the training data, and the number of classes trained into the DNN. With OpenMP, multicore, and GPU acceleration, the time required to interpret the computer screen with AVRA did not scale as the number of DNNs increased. With one DNN the latency from changes on the screen to updates in the GUI was approximately 40 seconds. Execution time scaled linearly so that 10 DNNs required 400s to analyze the screen and update the GUI. This challenge led to a change in approach, leaving

 ISBN: 978-1-61208-577-7

behind the slow but precise DNNs as future work, and instead using the text filter as the main text classifier in AVRA.

3) Execution Time Measurement for AVRA: The design is shown in Figure 4, removing the DNNs and connecting the output of the text filter to the RS. The image capturing and OCR in AVRA were modified as well. Instead of capturing the OCR of the whole screen, the fullscreen image was processed into text in slices, with each slice processed in a separate thread. The advantage of this approach was much faster OCR, but the downside was that this approach would miss text sliced at the lines between slices where the image was cut. To counter this, a second set of image slices offset by half of the slice height was also processed by OCR as well. This ensured that no onscreen text was missed by the OCR process. Figure 5 shows how a screen can be cut into $2n-1$ slices. Each thread is responsible for extracting text from its image slice and uploading the text to the server. Testing various settings for n yielded that $n=6$ had the lowest execution time on average for a computer screen of size $4800x3600$ pixels using an Intel Core i7 3.6GHz CPU, 16GB DDR3 RAM, an SSD hard disk, and 2 GeForce GTX 770 GPUs. For 5 trials with fullscreen color images, $n=6$ (11 image slices) led to an OCR execution time for a given slice of 4.4 ± 1.5 seconds per slice, and an overall OCR execution time of 8.5 ± 1.4 seconds per image. For 5 trials with fullscreen grayscale images, $n=6$ (11 image slices) led to an execution time of 2.9 ± 0.5 seconds per slice, and an overall execution time of 5.4 ± 0.9 seconds per image. The advantage of using color images was increased CNN precision and recall. To balance the response time and scalability with high recall, AVRA was set to process color images to maintain the recall level at the expense of execution time.

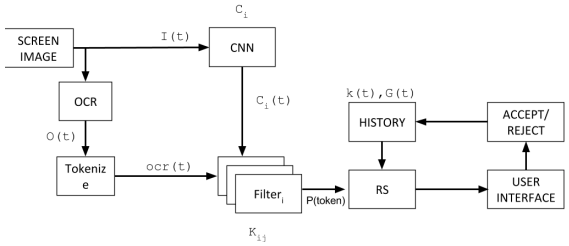

Figure 4. AVRA System Overview, after removing DNN to reduce execution time.

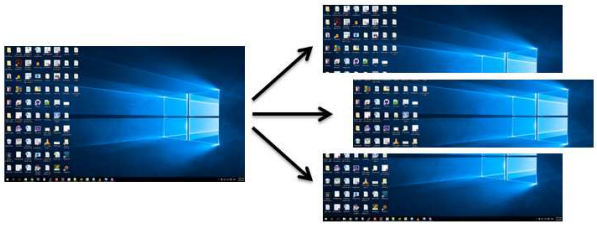

Figure 5. OCR capture process, splitting fulscreen images into $2n-1$ slices for executing overlapping subtasks in parallel threads to reduce OCR execution time.

Consider an example where AVRA is trained to recognize 11 contexts containing 2,103 $keywords$ overall. Tracking the flow of information through the color image processing design described above, and rounding to the nearest second, an image captured at time 0 passed enough information for the CNN to complete context recognition after 4 seconds, and the first context-filtered OCR results emerge from the text filter one second later. The RS results were available 4 seconds later, with another 2 seconds required to update the GUI. The minimum time between information appearing onscreen and a recommendation displaying on the GUI was therefore 11 seconds.

Improving on this design, the RS and filter were moved into the same file, obviating the database communication between these two modules. Tracking the flow of information through the new design and again rounding to the nearest second, an image captured at time 0 passed enough information for the CNN to complete context recognition after 4 seconds, and the first context-filtered OCR results emerged from the text filter combined with the RS after 2 seconds, with an average of less than one second required to update the GUI. The minimum time between information appearing onscreen and a recommendation displaying on the GUI was therefore 6 seconds.

The end-to-end execution time from detection onscreen to recommendation in a button was measured. To collect data from AVRA, the search text "looking for restaurant English text" was inserted into the Google Images API along with the restriction that the dimensions of the image results be exactly $1080x1920$ pixels, corresponding to the dimensions of the mobile phone used for testing. The first 50 results were saved as a dataset of images referred to as $SMALL\_IMAGES$.

To simulate the latency of image capture, a region of the desktop $1080x1920$ pixels was captured into a file for each processed image. After this simulated image capture delay, the CNN and OCR processes of AVRA were passed one of the static images from $SMALL\_IMAGES$. The total execution time required to fully process all 50 images of $SMALL\_IMAGES$ through the OCR, CNN, text filter, RS, and GUI was 176.0 seconds. The average execution time per image was 3.52 ± 1.51 seconds.

To ensure that AVRA can execute relatively quickly when many contexts have been trained into the CNN, ten latency samples were recorded for AVRA CNNs trained with 5, 50, 100, 200, and 400 contexts. Each sample was obtained by recording the CNN output after processing an image with dimensions $4800x3600$ pixels using a the same Intel Core i7-based system described above. The results, shown in Figure 6, reveal that the execution time grows with the number of added contexts. However, the incremental cost of adding contexts decreases with the number of contexts added, as shown in Figure 7. Furthermore, the execution time at 400 contexts remained low, at 3.97 ± 0.10 seconds. In Figure 6, the vertical axis shows the execution time required to process one $4800x3600$ pixel image using AVRA's CNN. Results for 5, 50, 100, 200, and 400 contexts are shown with error bars indicating the standard deviation for each measurement. A moving average line is added to the figure, revealing with a decreasing slope that the latency cost of adding more contexts is decreasing. In Figure 7, the vertical axis shows the execution time required to process one image, divided by the number of contexts trained into the CNN. Results for 5, 50, 100, 200, and 400 contexts are shown.

4) Recall, Precision, Precision-Recall and ROC Curve: It is interesting to observe the capabilities of AVRA regarding overlapping images because a program like AVRA must recognize onscreen items such as program windows that may overlap

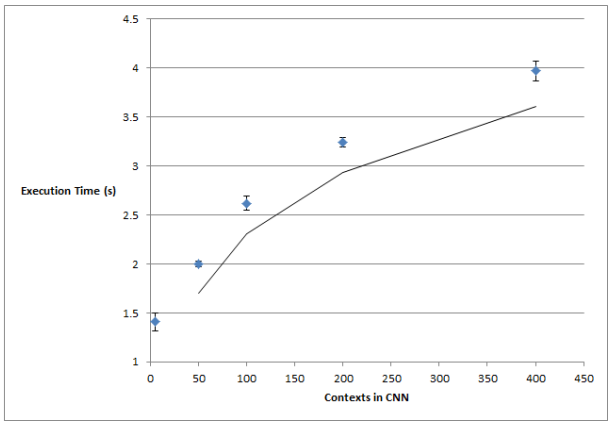

Figure 6. CNN latency as number of context increases.

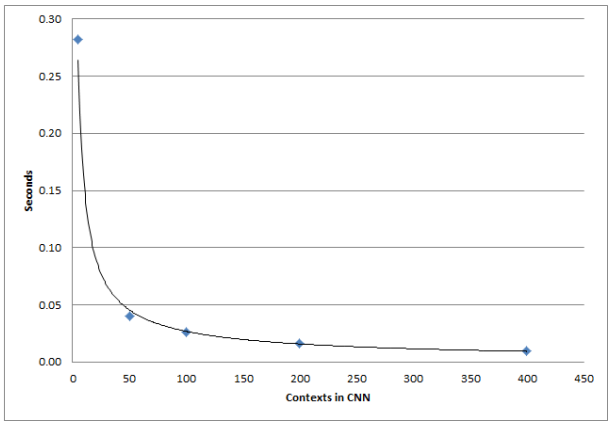

Figure 7. CNN latency per context.

when observed. AVRA recognizes multiple items in one image. For example, as shown in Figure 8, AVRA analyzing an image containing both a terminal window and an Eclipse IDE window leads to the recognition of both by the CNN. AVRA sometimes recognizes both contexts when they are partially occluded, but when overlap is high the occluded context (eclipse) was not recognized. However, in some cases the occluded context (in this case a terminal window) was recognized by the CNN. When two windows appeared side by side, the CNN usually recognized the context associated with each one.

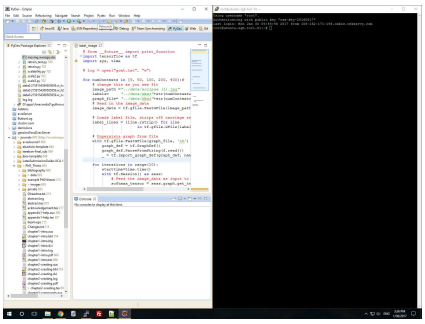

Figure 8. AVRA's CNN recognizing content in side by side windows. CNN output score was: eclipse (79.64%), console (12.73%), facebook (3.74%).

To observe the precision and recall for AVRA, it was tested by displaying to the system images on the full screen area one at a time. There were 50 $4800x3600$ pixel fullscreen desktop images, which contained one of two contexts (eclipse or console) and one specific $keyword$ known to the CNN for that context. The images were collected from real examples, and

TABLE V. CONFUSION MATRIX FOR ASSESSING AVRA.

		Predicted		Recall
		$C_A \wedge K_B$	$\neg C_A \vee \neg K_B$	
Actual	$C_A \wedge K_B$	36	14	0.7200
	$\neg C_A \vee \neg K_B$	0	50	
Precision		1.0000		

so they sometimes contained several other $keywords$ trained into AVRA for the context in question. For the eclipse context, the $keyword$ was `import`, and for the console context the $keyword$ was `apt-get`. Note that in eclipse the text is blue on a white background, and in a terminal the text is white on a black background. Neither of these scenarios is the typical OCR situation of black text on a white background. These examples were selected because they are a more realistic sample of text on a desktop computer than the obvious case of black text on a white background. To test for TN results 50 additional images containing no relevant context or keyword in them were also presented to AVRA one at a time. These 50 $4800x3600$ images were collected from Google Images using the keywords "my pictures" and the photo type filter was set to "photo". No FP examples were recorded for AVRA during the experiment, as the chance that a context and keyword are incorrectly selected by AVRA at the same time is very small.

The confusion matrix for the recorded observations is presented in Table V, after AVRA was trained on 11 classes. Samples either contained context C_A and keyword K_B, or they did not ($\neg C_A \vee \neg K_B$). The testing data had 25 images per class, with 1 class per testing image, one relevant $keyword$ per image, and classification threshold K=1% To generate an ROC curve for the collected AVRA data, a binary classification measure was used. Any sample with a TP results was associated with the ground truth state $[1, 0]$, and any sample with no TP results was associated with the ground truth state $[0, 1]$. The classifier guess was set to $[1, 0]$ in cases where TP or FP was observed. Otherwise the classifier guess for the sample was set to the state $[0, 1]$. The ground truth and classifier guesses were used to create the ROC curve of Figure 9, and the Precision-Recall curve of Figure 10. In 5 of the 50 images containing recommendable information, the recommendation was picked up by the text filter but was eliminated by the RS and 3 or more items the RS could recommend ranked higher than the $keyword$ of interest.

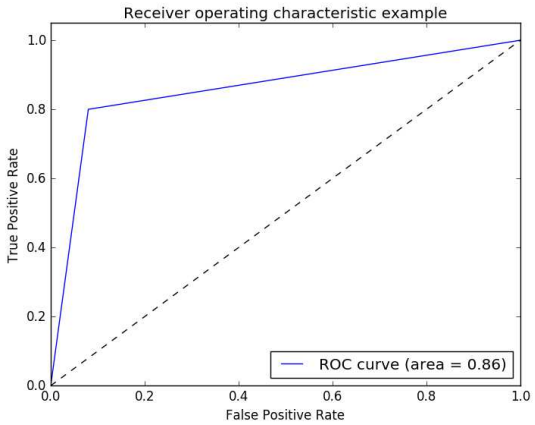

Figure 9. ROC curve for AVRA binary classification experiment

The following is an example of the weaknesses of AVRA's

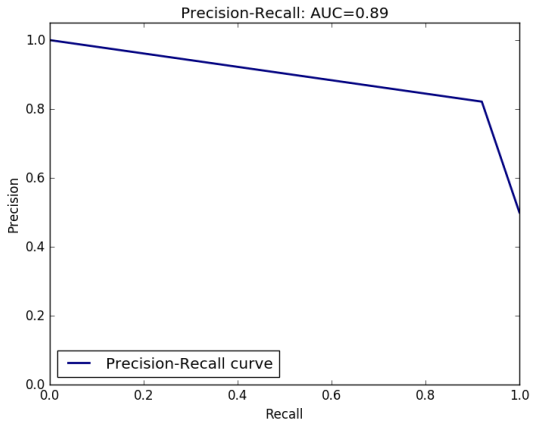

Figure 10. Precision-Recall curve for AVRA binary classification experiment

strict unsupervised learning approach described above. Due to the strict nature of the learning algorithm, when AVRA observes the user subtracting numbers in a calculator, the exact sequence of operations (e.g., $25,000 * 4$) is stored as a pattern to be recommended later on, rather than being learned symbolically as the multiplication of two numbers. Another limitation of the current approach to unsupervised learning is the keyword lexicon. At this time, only keywords already present in the word embedding model can be added as keywords in one of AVRA's DNNs. AVRA sometimes misses the context or keyword information, or has higher confidence in unhelpful recommendations than detected helpful recommendations. Two overall challenges in developing AVRA were poor classification of keywords with very short text length (e.g., the terminal command "ls"), and low context detection confidence (e.g., 2% confidence in the correct class). These cases were rare but noticeable. Perhaps the short keyword recognition could be resolved by modifying the DNN input filter hyperparameters. The low confidence context detection cases may be mitigated by collecting additional image data for context training.

VI. Conclusion

This work presented the overall design of AVRA, a virtual assistant for personal computer users. Measures of performance and measures of effectiveness were defined, and the design of AVRA's Mixed Initiative qualities was discussed. The performance of AVRA and a similar closed source commercial tool were recorded, and these two tools were compared in the context of the aforementioned requirements. The thesis statement claimed that a deep learning artificial intelligence can provide action recommendations related to onscreen messages. This work explained at a high level how action recommendations can be provided within a reasonable response time, and how these recommendations can be acted upon with a single mouse click.

An architecture for a deep learning recommender system for personal computer users was described in this work. Action recommendations produced by this design are personalized to the user and are generated in real-time. The AVRA system mines information from screen capture data, rather than interface with individual applications. Recommendations are presented to the user in an intuitive button-based user interface. The architecture described in this work can provide the foundation for further research into recommender system for personal computer users.

Future work planned for AVRA includes collaborative filtering and related privacy considerations, the expansion of AVRA's input processing and modeling capabilities, and unsupervised learning. Applying content-based image recognition and semantic segmentation of images for face and object classification within a context (and generating related recommendations) is an interesting area to explore as well.

References

[1] Amazon.com, Inc., "Alexa," https://developer.amazon.com/public/solutions/alexa, [retrieved: 2016-07].

[2] IBM, "IBM Watson Developer Community," https://developer.ibm.com/watson/, [retrieved: 2016-07].

[3] A. Smith, "Kite - Programming copilot," https://kite.com/, 2016, [retrieved:2016-05].

[4] Apple Inc., "iOS - Siri - Apple (CA)," http://www.apple.com/ca/ios/siri/, [retrieved: 2016-07].

[5] D. Kittlaus, "The team behind Siri debuts its next-gen AI Viv at Disrupt NY 2016," https://www.youtube.com/watch?v=MI07aeZqeco, 2016, [retrieved: 2016-07].

[6] V. Labs, "Viv," http://viv.ai/, [retrieved: 2016-07].

[7] SoundHound Inc., "SoundHound - Hound App," http://www.soundhound.com/hound, [retrieved: 2016-07].

[8] API.ai, "Assistant by Api.ai - Assistant.ai," https://assistant.ai/, [retrieved: 2016-07].

[9] N. Satt, "Hands-on with facebook m: the virtual assistant with a (real) human touch," http://www.theverge.com/2015/10/26/9605526/facebook-m-hands-on-personal-assistant-ai, 2015, [retrieved: 2016-07].

[10] J. Hempel, "Facebook launches m, its bold answer to siri and cortana," http://www.wired.com/2015/08/facebook-launches-m-new-kind-virtual-assistant/, 2015, [retrieved: 2016-07].

[11] K. Bell, "Facebook messenger chief: It will be years before everyone has m," http://mashable.com/2016/04/18/facebook-m-not-ready-for-years/#rnO6Nl3m1sqg, 2016, [retrieved: 2016-07].

[12] Y. LeCun, "EmTech MIT 2015: Augmented Knowledge Teaching Machines to Understand Us," https://youtu.be/a3DzL7Ad4PU?t=17m10s, 2015, [retrieved: 2016-07].

[13] Google, "Google Developers - Google Now," https://developers.google.com/schemas/now/cards, [retrieved: 2016-07].

[14] Q. Guo and Y. Song, "Large-scale analysis of viewing behavior: Towards measuring satisfaction with mobile proactive systems," in CIKM 2016, 2016.

[15] Microsoft, "Cortana - Meet your personal assistant - Microsoft - Global," https://www.microsoft.com/en/mobile/experiences/cortana/#, [retrieved: 2016-07].

[16] NextOS, "NextOS - Denise - Creating Virtual Life," http://www.nextos.com/, [retrieved: 2016-07].

[17] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in Proceedings of the 24th International Conference on World Wide Web. ACM, 2015, pp. 278–288.

[18] M. Lynlet, "Google unveils google assistant, a virtual assistant thats a big upgrade to google now," https://techcrunch.com/2016/05/18/google-unveils-google-assistant-a-big-upgrade-to-google-now/, 2016, [retrieved: 2016-07].

[19] J. Karlin, "Launchy," https://www.launchy.net/, 2016, [retrieved:2017-05].

[20] Google Inc, "Google Chrome," https://www.google.com/chrome/, 2016, [retrieved:2017-05].

[21] C. D. Wickens, H. Li, A. Santamaria, A. Sebok, and N. B. Sarter, "Stages and levels of automation: An integrated meta-analysis," in Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 54. SAGE Publications, 2010, pp. 389–393.

 ISBN: 978-1-61208-577-7

[22] S. Makonin, D. McVeigh, W. Stuerzlinger, K. Tran, and F. Popowich, "Mixed-initiative for big data: The intersection of human+ visual analytics+ prediction," in 2016 49th Hawaii International Conference on System Sciences (HICSS). IEEE, 2016, pp. 1427–1436.

[23] D. Shapiro, N. Japkowicz, M. Lemay, and M. Bolic, "Fuzzy string matching with character-level deep neural network classification," Applied Artificial Intelligence, submitted 2016.

[24] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, 2015, pp. 529–533.

[25] K. C.-C. Chang, B. He, and Z. Zhang, "Metaquerier over the deep web: Shallow integration across holistic sources," in In Proceedings of the VLDB Workshop on Information Integration on the Web, 2004.

[26] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," Communications of the ACM, vol. 40, no. 3, 1997, pp. 66–72.

[27] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," Artificial Intelligence Review, vol. 13, no. 5-6, 1999, pp. 393–408.

[28] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, Collaborative Filtering Recommender Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 291–324. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72079-9_9

[29] J. Gemmell, T. Schimoler, M. Ramezani, L. Christiansen, and B. Mobasher, "Resource recommendation for social tagging: a multi-channel hybrid approach," Recommender Systems & the Social Web, Barcelona, Spain, 2010.

[30] S. Maneeroj and A. Takasu, "Hybrid recommender system using latent features," in Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on. IEEE, 2009, pp. 661–666.

[31] L. T. Weng, Y. Xu, Y. Li, and R. Nayak, "Exploiting item taxonomy for solving cold-start problem in recommendation making," in 2008 20th IEEE International Conference on Tools with Artificial Intelligence, vol. 2, Nov 2008, pp. 113–120.

[32] L. Martinez, L. G. Perez, and M. J. Barranco, "Incomplete preference relations to smooth out the cold-start in collaborative recommender systems," in Fuzzy Information Processing Society, 2009. NAFIPS 2009. Annual Meeting of the North American. IEEE, 2009, pp. 1–6.

[33] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, and E. Duval, "Context-aware recommender systems for learning: a survey and future challenges," IEEE Transactions on Learning Technologies, vol. 5, no. 4, 2012, pp. 318–335.

[34] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in Recommender systems handbook. Springer, 2011, pp. 217–253.

[35] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," Knowledge-Based Systems, vol. 46, 2013, pp. 109–132.

[36] N. Sawadsky, G. C. Murphy, and R. Jiresal, "Reverb: Recommending code-related web pages," in Proceedings of the 2013 International Conference on Software Engineering. IEEE Press, 2013, pp. 812–821.

[37] N. Sawadsky and G. C. Murphy, "Fishtail: from task context to source code examples," in Proceedings of the 1st Workshop on Developing Tools as Plug-ins. ACM, 2011, pp. 48–51.

[38] E. Murphy-Hill and G. C. Murphy, "Recommendation delivery," in Recommendation Systems in Software Engineering. Springer, 2014, pp. 223–242.

[39] O. Madani, H. H. Bui, and E. Yeh, "Efficient online learning and prediction of users' desktop actions." in IJCAI, 2009, pp. 1457–1462.

[40] B. Korvemaker and R. Greiner, "Predicting unix command lines: adjusting to user patterns," in AAAI/IAAI, 2000, pp. 230–235.

[41] L. B. Holder and D. J. Cook, "Automated activity-aware prompting for activity initiation," Gerontechnology: international journal on the fundamental aspects of technology to serve the ageing society, vol. 11, no. 4, 2013, p. 534.

[42] Y.-a. Kang and J. Stasko, "Characterizing the intelligence analysis process: Informing visual analytics design through a longitudinal field study," in Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on. IEEE, 2011, pp. 21–30.

[43] P. Zhang and D. Soergel, "Towards a comprehensive model of the cognitive process and mechanisms of individual sensemaking," Journal of the Association for Information Science and Technology, vol. 65, no. 9, 2014, pp. 1733–1756.

[44] J. Allen, C. I. Guinn, and E. Horvtz, "Mixed-initiative interaction," IEEE Intelligent Systems and their Applications, vol. 14, no. 5, 1999, pp. 14–23.

[45] K. Cook, N. Cramer, D. Israel, M. Wolverton, J. Bruce, R. Burtner, and A. Endert, "Mixed-initiative visual analytics using task-driven recommendations," in Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on. IEEE, 2015, pp. 9–16.

[46] P. Pu and D. Lalanne, "Design visual thinking tools for mixed initiative systems," in Proceedings of the 7th international conference on Intelligent user interfaces. ACM, 2002, pp. 119–126.

[47] G. Corrado, "Computer, respond to this email." https://research.googleblog.com/2015/11/computer-respond-to-this-email.html, November 2015, [retrieved: 2017-05].

[48] B. Schwartz, The paradox of choice: why more is less. New York: Ecco, 2004.

[49] Peter Evers, "images-scraper," https://github.com/pevers/images-scraper, 2016, [retrieved: 2016-07].

[50] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" CoRR, vol. abs/1411.1792, 2014. [Online]. Available: http://arxiv.org/abs/1411.1792

[51] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[52] B. Chapman, G. Jost, and R. Van Der Pas, Using OpenMP: portable shared memory parallel programming. MIT press, 2008, vol. 10.

[53] LISA lab, "Multi cores support in Theano," http://deeplearning.net/software/theano/tutorial/multi_cores.html, [retrieved: 2016-07].

[54] J. L. Hennessy and D. A. Patterson, Computer architecture: a quantitative approach. Elsevier, 2011.

[55] M. Beissinger, "How to install Theano on Amazon EC2 GPU instances for deep learning," http://markus.com/install-theano-on-aws/, 2015, [retrieved: 2016-07].

[56] NVIDIA Corporation, "CUDA FAQ — NVIDIA Developer," https://developer.nvidia.com/cuda-faq, [retrieved: 2016-07].

 ISBN: 978-1-61208-577-7

A Data Modelling and Visual Analysis of Science and Technology Policy Keyword

Seung Su Chun
Division of Information & Knowledge
Korea Institute of Science and Technology Evaluation and Planning
Seoul, South Korea
dabins@kistep.re.kr

***Abstract*— Recently, enterprises and major countries are increasing their R&D investment for economic growth through science and technology upbringing. As the size of investment increases, it becomes more difficult to understand and analyze the status and performance of detailed projects, programs and technologies. This paper deals with a method of analyzing the relationship between major technological keywords by text mining large scale policy documents for R&D investment. This methodology helps us better understand the relationship between the complicated policy/technology and the investment flow, and support the analysis with a visualization of the relationship.**

Keywords-technology policy; data mining; network model; visual analysis

I. DATA PROPERTY OF SCIENCE AND TECHNOLOGY POLICY ANAYSIS

In today's R&D investment, it is essential to understand the relationship between investment structure and technology. Analysis of investment policy in research and development is essential for understanding the relationship between detailed programs and technology. In addition, to coordinate investment in R&D, it is necessary to understand the relationship between technologies as well as investment flows. Recently, a network analysis methodology that can grasp the relationships between various entities through intuitive visualization and apply sophisticated analytical methods has been suggested as a useful approach. Network analysis is one of the Business Intelligence analytical methods for strategy establishment, rapid decision making and risk management. It can model meaningful data by networking complicated data. The development in technology and the maturity of an information-knowledge based society accelerate technological competition among enterprises and countries [1]. These sophisticated systems are designed to find useful information from the large data repositories and developed to be used quickly in decision making. The availability of data, such as big data systems, has become widespread, and researches have been actively conducted on visualization techniques for computational thinking and rapid decision support [2]. The progress of intelligent systems through relational meaning and interpretation by structuring and modeling analysis of data and information is being promoted in various fields.

Here, we have demonstrated the application of data structuring methods and visualization techniques for the analysis of unstructured data statistics on R&D policies.

II. VISUAL ANALYSIS OF NETWORK MODELING

In the past days, investment analysis used to be based on business classification. But it is important to analyze the linkage between business and technology in convergence-oriented R&D, and the visual analysis becomes very useful. Network modeling is used as a key process for network analysis visualization by conceptually defining the entities that are components of a network map and the relationships between them. The action of policy production, refinement, evidence, and diffusion can be seen as a composition of content types such as knowledge, people, and activity. Complex policy documents can be classified into people, services, processes and resources, and reconstructed into hierarchical networks [3]. Firstly, we have organized the policy document as in Table 1 and extracted the keywords from the contents of the document by the text mining technique. We have visualized the relationship between keywords as shown in Figure 1 by analyzing the proximity (distance) and relevance (frequency) between the keywords in the context and sentences in the document.

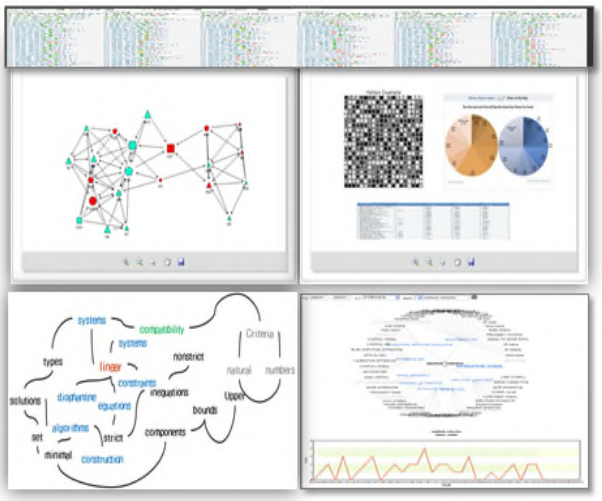

Figure 1. Example of keyword based network analysis

 ISBN: 978-1-61208-577-7

TABLE I. META MATRIX OF POLICY KEYWORD

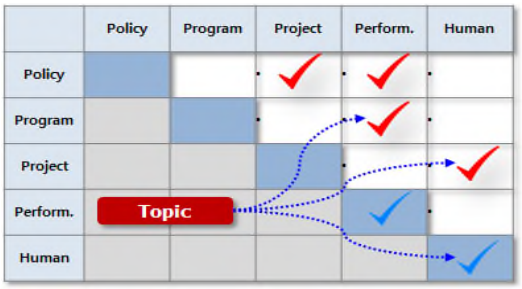

Table 1 shows the document classification system for R&D investment, which is used as the attribute value of keyword and relation. The document classification in Table 1, such as the policy decision process, is used to analyze the frequency changes and topic relationships of specific keyword topics extracted through text mining. The data of keywords identified as entities (nodes, points) that can generate a network map can be classified into five types: policy, program, project, performance, and human. The extracted keywords can represent the relationship between the categories with the document classification of the meta-metrics as an attribute. The same keywords extracted from the meta matrix documents express the relationship between different categories and visualize them by constructing main topics between the categories [4][5]. The keyword based network model is important to understand the decision flow for a specific technology because decision-making in R&D investment, establishment of strategies, and evaluation and adjustment of R&D programs are all sequentially processed.

III. POLICY TOPIC DETECTION AND MODEL ANALYSIS

The aim of this paper is to analyze and visualize the analytical relationship between keywords using the existing business classification method, which employs text mining. This section describes how to calculate the relationship between policy documents and keyword topics. In order to understand the implied meaning of knowledge, the interaction and effects between topics should be analyzed in-depth as they are the meaning unit of knowledge [6]. In the analysis of the relationship between keywords, we use the Celton index method for properties such as connectivity between nodes and proximity. In this study, we define the process (*P*) document of classification as Table 1 and analyze the relevance of the extracted keywords for the Selton index. The keyword-based network model was used for efficiency analysis of the technology topic. At first, the intensity of relations may be quantified based on the connection intensity between certain topics. P_i is the number of i in a certain topic, and P_j is the number of j in a certain topic, and P_{ij} is the number of i and j in a certain topic. If the intensity of relations is r, and r is calculated as the distance between the context and the frequency of appearing at the same time between different topics, then:

$$r = \frac{P_{ij}}{\sqrt{P_i P_j}}$$

The connection relations are divided into degree, closeness and proximity, and have absolute value and relative value. The Degree Centrality (C) of having the most nodes between topics has a higher connection intensity when certain topics are revealed in other major areas of knowledge, and it has a central role in this area. Distance refers to the inter-context distance between any keyword topics. The Closeness Centrality has a central role when certain topics are close to other topics, and distance (D) D_{ij} is the shortest course which connects between topics i and j, and g is the number of the overall node. In the network, the extracted keyword topic is a node. When the relation value r between the topics is high, the edge is constituted. The Between Centrality, where different networks are connected, plays a mediating role in the relation and is necessary for interpreting relationships of different topics. g_{jk} is the shortest course number between two topics (j and k), $g_{jk}(i)$ is the frequency of passing topic i between two topics j and k($j \neq k$), and g is the number of the overall node. Through centrality analysis between topics, we can interpret meaningful characteristics of the knowledge model, and expand the range with difference and connection between knowledge models.

IV. CONCLUSIONS

In this paper, we have described the method of visualizing the relationship of keywords extracted through text mining in the policy document for technology investment and analyzing them by network. This topic network visualization helps to understand the relationships between specific keywords easily and helps interpret the structure of relations with document classification according to the policy process. We have also demonstrated a method for interpreting the meaning according to the network structure and the distance between keyword topics. In future work, we will apply this technique to a large amount of data and complex documents, and also develop a tool to visually analyze the relational path between specific keyword topics.

REFERENCES

[1] G. Yezersky, “General Theory of Innovation, International Federation for Information Processing,” Vol.250, Springer, 2007, pp. 45-55.

[2] N. L. Rovira, “the future of computer aided innovation, IFIP International Federation for Information Processing,” Springer, Vol.277, 2008, pp. 3-4.

[3] C. Heitmeyer, “Using abstraction and model checking to detect safety violations in requirements specifications,” IEEE Transactions on Software Engineering, vol. 24, no. 11, 1998.

[4] W. Chan, “Temporal Logic Queries,” Proceedings of CAV 2000, Springer, LNCS 1855, 2000.

[5] X. Wang and T. Vitvar, “WSMO-PA: Formal Specification of Public administration service model on Semantic Web Service Ontology,” IEEE ICSS’07, 2007.

[6] W3C, “Web Service Description Language (WSDL),” ver.2.0 recommendation, 2007.

 ISBN: 978-1-61208-577-7

Approximate Convex Decomposition for Real-time Terrain Fracturing

Namil Lee
Department of Computer Science and Engineering
Korea University
Seoul, Korea
e-mail: 2namil@naver.com

JungHyun Han
Department of Computer Science and Engineering
Korea University
Seoul, Korea
e-mail: jhan@korea.ac.kr

***Abstract*— This paper proposes the Terrain Approximate Convex Decomposition (TACD) method. It extracts features from the regularly-sampled height map, uses the features to decompose the height map into a set of convex hulls, and then applies the fracture patterns to the convex hulls. The TACD method runs at real time and presents realistic effects. It can be integrated with games and virtual reality applications.**

Keywords-terrain; height map; fracturing.

I. INTRODUCTION

Fracturing or fragmentation refers to the process by which an object is shattered into small pieces. It is a desired effect especially in games, but real-time terrain fracturing is not easy to implement.

It is generally required to use pre-defined *fracture patterns* for real-time fracturing, but the terrain area is basically a 2D surface and is too wide to apply such a pattern. This paper tackles the problem by decomposing the input terrain surface into 3D convex hulls, each of which is independently combined with a fracture pattern.

This paper presents the real-time terrain fracturing algorithms in the following organization: Section II presents the basics of pattern-based fracturing. Section III proposes what we call Terrain Approximate Convex Decomposition method. Section IV presents the experimental results and Section V concludes the paper.

II. CONVEX-HULL FRACTURING

This section first presents how fracturing is made with a pattern, and then how the terrain is fractured in terms of features and convex hulls.

A. Pattern-based Fracturing

For real-time fracturing, we use a pre-defined fracture pattern. Figure 1 shows the flow of fracturing. The input object is decomposed into convex hulls, to which the fracture pattern is applied so that the intersection of the convex hulls and fracture pattern produces many facture cells. The cells will be physically simulated to bring the fracturing effects.

B. Feature Extraction

The most commonly used terrain model is the *height map*, which stores discrete elevation data on the grid. The first step of terrain fracturing is to extract features from the height map.

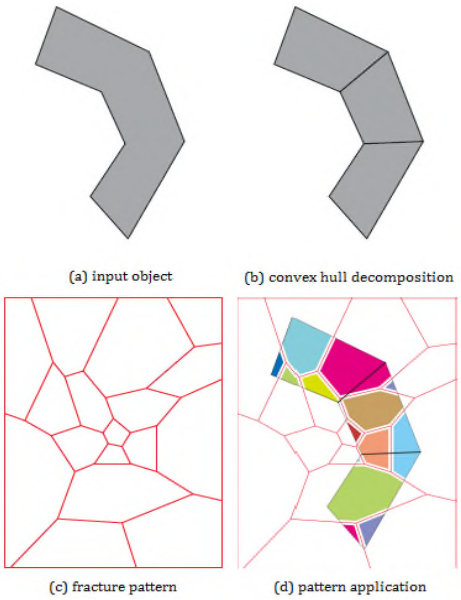

Figure 1. Fracturing flow.

In our study, we used a 256×256-resolution height map. Its base is the *xz*-plane with *y*-coordinate zero, and the terrain surface is a mesh of 3D triangles constructed with 256^2 vertices. Figure 2-(a) shows a cross section of an example height map, which was simplified to have 11×11 samples just for visualization purpose. The red and green points represent the features. The red-colored features are extracted through the Laplacian operation on the surface curvatures of the height map. The green-colored features represent the boundary of the 256×256-resolution height map volume.

C. Convex Hull Decomposition

If we computed the convex hull for the entire height map shown in Figure 2-(a), we would have a big convex hull, as illustrated with the red boundary. It is not efficient to apply the fracture pattern to such a big convex hull.

In order to decompose the height map into a set of smaller convex hulls, we need to extract another group of features, such as the red point in Figure 2-(b). It represents a point with the locally-maximal concavity. A wide height map may contain a number of such features, and they are

 ISBN: 978-1-61208-577-7

extracted through flooding-based watershed technique [1]. Figure 3 illustrates the process.

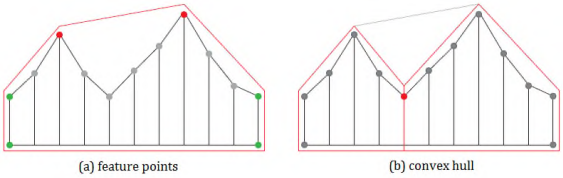

Figure 2. Features in the height map.

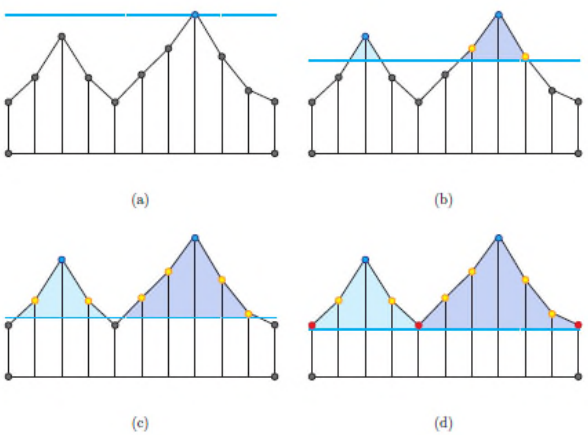

Figure 3. Flooding-based watershed.

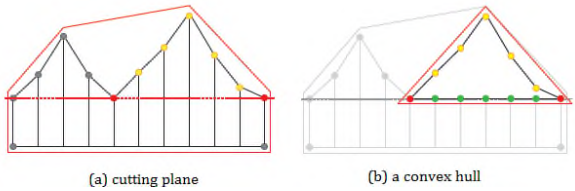

Figure 4. Convex hull computation.

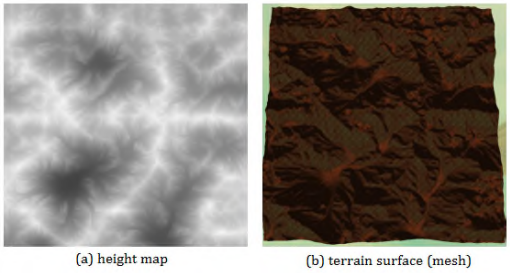

Figure 5. Height map and terrain surface.

Given a set of four or more nearby feature points, a *cutting plane* is defined, as shown in Figure 4-(a). Then, a convex hull is obtained with the cutting plane, as illustrated in Figure 4-(b). In the current implementation, we use the Random Sample Consensus (RANSAC) method [2] for defining the plane and the Quickhull method for computing the convex hull [3].

III. EXPERIMENTAL RESULTS

The proposed system was implemented in Direct3D 11 on Intel i7-3770 3.40GHz CPU, 16GB RAM, and GeForce GTX 980 Ti.

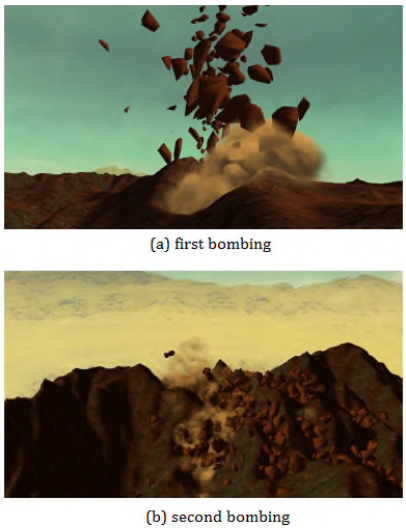

Figure 6. Terrain fracturing due to bombing.

Figure 5 shows the height map of resolution 256^2 and the 3D polygon mesh constructed from it. Figure 6 shows the fracturing effects. The low-frequency terrain in Figure 6-(a) produces 142 fracture cells, and fracturing takes 28.6ms. The high-frequency one in Figure 6-(b) produces 205 fractures, and fracturing takes 30.63ms. The higher-frequency terrain surface produces more fractures, which consume more time.

IV. CONCLUSION

This paper presented a novel terrain fracturing method based on fracture pattern, where the features are extracted from the height map, and then the height map is decomposed into convex hulls using the features. The fracture patterns apply to the convex hulls. The method can be integrated with games and virtual reality applications.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.R0115-16-1011). J. Han is the corresponding author.

REFERENCES

[1] E. Zuckerberger, A. Tal, and S. Shlafman, "Polyhedral surface decomposition with applications," *Computers & Graphics*, vol. 26, no. 5, 2002, pp.733 –743.

[2] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, 1981, pp.381–395.

 ISBN: 978-1-61208-577-7

[3] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, “The quickhull algorithm for convex hulls,” ACM Trans. Math. Software, vol. 22, no. 4, 1996, pp. 469–483.

 ISBN: 978-1-61208-577-7

Efficient Hash Generation Method for Intra Block Copy Search in HEVC Screen Content Coding

Ilseung Kim
Department of Electronics and Computer Engineering
Hanyang University
Seoul, Korea
e-mail: ghanjang@gmail.com

Jechang Jeong
Department of Electronics and Computer Engineering
Hanyang University
Seoul, Korea
e-mail: jjeong@hanyang.ac.kr

***Abstract*— Recently, the standardization of HEVC-SCC was finalized, including a hash-based intra block copy (IBC) search scheme which brings significant coding gains to code screen content. However, the hash table generation itself creates a massive computational burden on the encoder side. In this paper, we propose a picture-wise hash generation method to reduce significantly the complexity of hash-based IBC search. Experimental results show that the proposed scheme reduces about 77% of hash generating time, which leads to 43% total IBC searching time saving in all-intra (AI) coding structures, compared with the hash-based IBC search in the HEVC-SCC test model (SCM)-8.0.**

Keywords-HEVC; Screen Content Coding (SCC); Hash; Intra Block Copy (IBC).

I. INTRODUCTION

High Efficiency Video Coding standard (HEVC) [1] is the state-of-the-art video coding standard and it was finalized in January 2013. HEVC doubles the coding efficiency of video compression compared with H.264/MPEG-4 AVC standard under the equivalent subjective visual quality circumstances [2][3].

Recently, on the other hand, computer generated content, called screen content, has surged with the rapid development of multimedia techniques and applications, such as wireless display, remote desktop, external display interfacing, and cloud computing, etc. [4]. Requests have been made in asking for investigation of new coding tools for screen content [4]. In order to reflect these requests, the MPEG requirements subgroup published a set of requirements for an extension of HEVC for coding of screen content in January 2014 [4] and currently the HEVC Screen Content Coding (SCC) Draft 6 was published in February 2016 [5][6].

Different from the traditional camera-captured content, screen content often has mixed content consisting of rendered graphics, text, or animation as well as camera-captured content in one picture. In addition, as shown in Fig. 1, screen content has different characteristics compared with that of the camera-captured content, such as containing no sensor noise, having large uniformly flat areas, repeated patterns, and a limited number of different colors and so on.

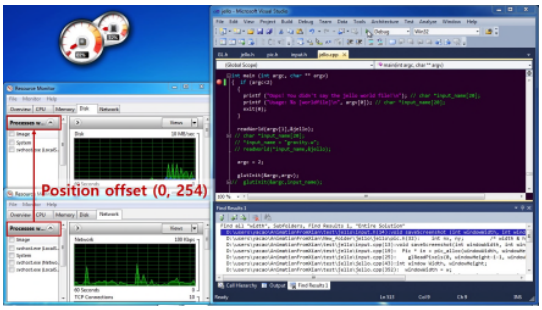

Figure 1. Example of the long-distance repeated patterns of the sequence 'sc-programming'.

In order to improve the coding efficiency for screen content doing, HEVC-SCC has developed many new tools. Palette mode utilizes the observation that a limited number of different color values frequently exist for screen content [7]. A lot of HEVC-SCC test materials consist of RGB colour format or YCbCr 4:4:4 format, whose inter-colour component correlation is very high. In order to remove inter-colour component redundancy, adaptive color transform (ACT) has been introduced [8]. Unlike camera- captured content, there is no need to use fractional motion compensation for much screen content. For this reason, adaptive motion vector resolution (AMVR) has adopted in SCC [9]. There are many repeated patterns, such as characters in screen content, so the motion estimation and compensation within the current picture can be effective. Intra block copy (IBC) can get a good prediction from the block with the same pattern in the coded region on the current frame [10].

Long range repeated patterns and large motions are often observed in screen contents as shown in Fig. 1. In order to support large range block search without bringing heavy computation burden, the hash-based global IBC search was adopted in HEVC-SCC draft 6 [5]. To perform the hash-based IBC search, hash tables of the reconstructed coding tree unit (CTU) are generated first. The hash value as well as the block position of each block are calculated and stored in the hash table. The blocks sharing the same hash values with the current coding block will be selected as candidates and rate distortion optimization (RDO) based block matching is

 ISBN: 978-1-61208-577-7

conducted with the candidates. Detailed process of the hash-based IBC search will be introduced in Section II. However, hash table generation itself still has heavy computation burden. In this paper, we present an efficient way to generate hash values for hash-based IBC search in HEVC-SCC.

This paper is organized as follows: Section II provides an overview of technical issues of IBC and presents a tool for improving IBC, especially hash-based block vector search as a conventional algorithm. Section III presents the proposed algorithm. Section IV discusses the experiment results and the conclusion is set forth in Section V.

II. INTRA BLOCK COPY (IBC) IN HEVC-SCC

Basically, IBC is one of the block matching techniques with the reconstructed current picture before loop filtering. In other words, IBC is the technique that searches the block vectors (BVs) in terms of the lowest rate-distortion (RD) cost within the reconstructed current picture. When calculating RD cost, chroma components are used as well as luma components. First, find the four best BVs by calculating RD cost of luma components and it can be expressed as follows:

$$\mathrm{RD}_{\mathrm{cost}} = SAD_{luma} + \lambda \times BV_{bits}. \quad (1)$$

where SAD_{luma} is sum of absolute differences of luma components, λ is a Lagrangian multiplier, and BV_{bits} is the number of bits needed to signal the BV, respectively. For the chosen four BVs, additional RD calculation process, called the chroma refinement process, is performed in order to find the optimal BVs and it can be expressed as followed:

$$\mathrm{RD}_{\mathrm{cost}} = SAD_{luma} + SAD_{chroma} + \lambda \times BV_{bits}. \quad (2)$$

where SAD_{chroma} is sum of absolute differences of chroma components. The SAD of two N x N blocks is defined as

$$SAD = \sum_{i,j=0}^{N} |f_c(i,j) - f_r(i-x, j-y)|. \quad (3)$$

where $f_c(i,j)$and $f_r(i,j)$denote the intensity of the pixel with coordinate (i, j) in the current picture and the reference picture, (x, y) represents the displacement of the BVs, respectively.

As aforementioned, local and global search methods are introduced in HEVC-SCC. Local searches are performed first and global searches are followed. Finally, decide the optimal BVs by comparing both searches in terms of RD cost. For the local search, only two CTUs, the current and the left CTU, are used as a search range as depicted in Fig. 2.

Global search is conducted only for 16x16 and 8x8 prediction unit (PU) and the entire reconstructed current picture before loop filtering can be a search range for global search. For 16x16 PU, a one-dimensional search method is used, which can search only one direction, horizontally or vertically as shown in Fig.3.

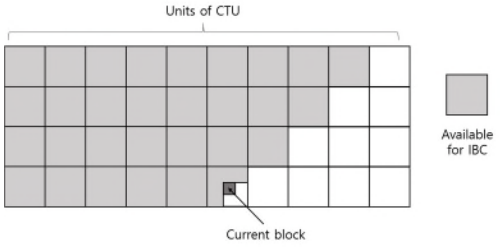

Figure 2. Global block vector search prediction area

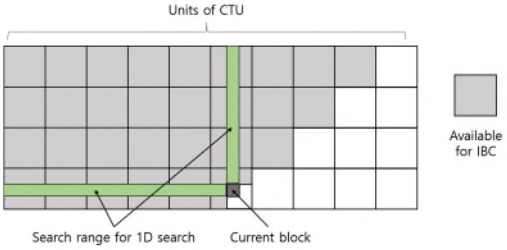

Figure 3. A one-dimensional search prediction area for 16x16 blocks

For 8x8 PU, a hash-based search is used for the entire picture but available. The 16-bit hash entries for the current block and the reference block are calculated using the original sample values. Let *Grad* denote the gradient of the 8x8 blocks
and *dc0*, *dc1*, *dc2*, and *dc3* denote the DC values of the four 4x4 sub-blocks of the 8x8 block. Then, the 16-bit hash entry *H* is calculated as:

$$\begin{aligned} H = {} & MSB(dc0,3) \ll 13 + MSB(dc1,3) \ll 10 \\ & + MSB(dc2,3) \ll 7 + MSB(dc3,3) \ll 4 \\ & + MSB(Grad,4) \end{aligned} \quad (5)$$

where MSB(X,n) represents the n most significant bits of X.

The procedure of hash-based IBC search is as follows. The values of H are calculated when the CTUs were reconstructed and they are stored in the hash list, unless the value of $Grad$ is less than 5. After calculating the value of H of the current PU, perform the RD process of the luma component using (1) for the blocks which have the same hash value with that of the current PU and choose the eight best BVs. For the chosen eight best BVs, additional RD cost is calculated using (2). By doing so, the optimal BV of the global search can be determined.

III. PROPOSED ALGORITHM

Hash-based IBC mode has a critical role in HEVC-SCC. Compared with SCC without hash-based IBC, up to 75% Bjøntegaard Distortion bitrate (BD-BR) gain can be achieved in our experiment. However, hash generation and hash table updating method in HEVC-SCC brings a lot of computational redundancy.

In HEVC-SCC, the number of search points N_{search_point} required to calculate hash values are

TABLE I. THE NUMBER OF OPERATIONS TO OBTAIN A HASH VALUE H PER BLOCK IN HEVC-SCC

	Number of operations	if $N = 8$
Addition	$4(N-1)(N-1)$	196
Shift	$(N-1)(N-1)+4$	53
Absolute	$2(N-1)(N-1)$	98

TABLE II. THE NUMBER OF OPERATIONS TO OBTAIN A HASH VALUE H PER BLOCK IN THE PROPOSED ALGORITHM

	Number of operations (if $N = 8$)
Addition	$\cong 15$
Shift	$\cong 1+4$
Absolute	$\cong 2$

$$N_{search\_point} = O\{(W_{pic}-N)(H_{pic}-N)+\left(\frac{W_{pic}H_{pic}}{N^2}\right)\} \quad (6)$$

where W_{pic} and H_{pic} are the picture width and height, N denotes block size, and $O\{\cdot\}$ represents a big-O notation, respectively. Moreover, $(N-1)^2$ addition operations and 4 shift operations for computing *dc* values, and $2(N-1)^2$ absolute operations, $(N-1)^2$ shift operations and $3(N-1)^2$ for calculating *Grad* value are required to calculate the hash value *H*. The number of operations required to obtain a hash value *H* per block is summarized in Table I.

However, the number of required operations to compute hash entry H is always same at every search point without considering the correlation among them, even though many operational overlaps occur between neighboring blocks. Moreover, updating the hash table right after the reconstruction of each CTU also causes unnecessary duplication of calculations. Hash entry *H* is calculated using the original sample values, not reconstructed values; thus, there is no need to update hash table after the reconstruction of CTUs.

In this paper, we propose a picture-wise hash generation method using integral image to effectively eliminate unnecessary operations for hash generation and hash table updating in HEVC-SCC.

Integral image is widely used to calculate the sum norm values, as shown in Fig. 4. The required operations for the value of *dc*s and *Grad* in eq. 4 to obtain the hash value *H* are closely related to the way to compute the sum-norm of the relevant block. Fig. 5 shows examples of required operations to obtain the sum norm of two consecutive $N \times N$ blocks. Compared to the method in HEVC-SCC which requires $(N-1)^2$ additional operations to calculate the sum norm value of each $N \times N$ block, only three addition operations are needed by using an integral image.

To calculate the integral image, the following number of addition operations $N_{addition}$ are required

$$N_{addition} = 3W_{pic}H_{pic} - 2W_{pic} - 2H_{pic} + 1. \quad (7)$$

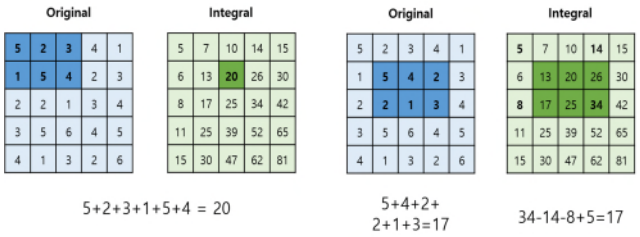

Figure 4. Example of an integral image.

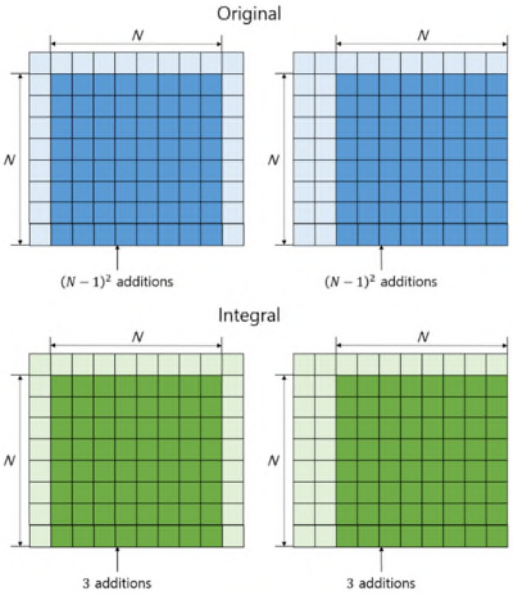

Figure 5. Examples of required operations to obtain the sum norm of two consecutive $N \times N$ blocks

Considering eq. 6, the number of operations required to make an integral image per search point are approximately 3 additions.

The procedure of the proposed algorithm is as follows: first, create the gradient map for the current picture. In this stage, $3(W_{pic}-1)(H_{pic}-1)$, $2(W_{pic}-1)(H_{pic}-1)$, and $(W_{pic}-1)(H_{pic}-1)$ of addition, absolute, and shift operations are required in a picture unit, respectively. Then, the remaining operations needed to acquire hash entry H are the same as those used to calculate dc values in eq.4, which are highly related to calculate the sum norm value of the relevant block. Generating an integral image of the gradient map for the current picture is followed by. Using the integral image of the gradient, the value of *Grad* in (4) can be calculate and then, stop hash generation process if the value of *Grad* is less than 5. After that, make an integral image of the current picture and calculate the DC values of the four 4x4 sub-blocks using the integral image of the current picture. By doing so, the hash values of all search points in

the current picture can be efficiently generated. The number of operations in the proposed algorithm is summarized in Table II and it shows the amount of the decreasing number of operations compared to that in HEVC-SCC as shown in Table II.

IV. EXPERIMENTAL RESULTS

The simulation was conducted in 4:4:4 test sequences under the common test conditions (CTC) used during the development of HEVC-SCC [6] under the intra main SCC configuration. In order to test the screen content in a variety of environments, HEVC-SCC CTC includes several types of sequences such as text and graphics with motion (TGM), mixed content (M), animation (A), and camera-captured content (CC). HM-16.8+SCM-8.0 was used as the anchor, and 22, 27, 32, and 37 are set for QPs.

Table III and IV showed the time comparison result for the hash generation and total IBC searches to verify computational complexity of the proposed algorithm. The total IBC searching time consisted of the hash generation time and the hash-based search time including RDO process for the candidates which matched the hash value with the that of the current block. In order to compare the consumed time, time rate was used with the following equation:

$$Time\_rate = \frac{time_{proposed}}{time_{anchor}} \times 100\%. \quad (6)$$

The results in Table III showed that the same results can be achieved by taking 21.83% on average of time compared to anchor when generating IBC hashes. Using the method in the proposed algorithm, the total IBC search time was up to 24.38% and 56.92% on average compared to that of anchor. Note that coding efficiency between the anchor and the proposed algorithm is equivalent, because every methods except hash generation method between them are same.

V. CONCLUSION

This paper proposes an efficient hash generation method for IBC search after analyzing the conventional IBC technique in HEVC-SCC draft 6, especially hash-based IBC. The proposed algorithm suggests an efficient way to generate hash values in picture units at once by using integral images and the early termination method. The experimental result shows that the hash generation time rate is approximately 23%, which leads to 43% total IBC searching time saving in all-intra (AI) coding structures, compared with the hash-based IBC search in the HEVC-SCC test model (SCM)-8.0.

TABLE III. HASH GENERATION TIME COMPARISON BETWEEN HM-16.8+SCM-8.0 AND THE PROPOSED ALGORITHM.

Category	Sequences	Time rate (%)
TGM, 1080p	sc_flyingGraphics	17.68
	sc_desktop	23.60
	sc_console	25.35
TGM, 720p	sc_web_browsing	21.82
	sc_map	22.71
	sc_programming	21.86
	sc_SlideShow	27.98
M, 1440p	Basketball Screen	21.83
	MissionControlClip2	23.50
M, 1080p	MissionControlClip3	21.72
A, 720p	sc_robot	24.00
CC, 1080p	Kimono1	24.88
Average		**23.08**

TABLE IV. TOTAL IBC SEARCHING TIME COMPARISON BETWEEN HM-16.8+SCM-8.0 AND THE PROPOSED ALGORITHM.

Category	Sequences	Time rate (%)
TGM, 1080p	sc_flyingGraphics	80.85
	sc_desktop	80.76
	sc_console	80.84
TGM, 720p	sc_web_browsing	50.52
	sc_map	58.21
	sc_programming	34.71
	sc_SlideShow	67.85
M, 1440p	Basketball Screen	24.38
	MissionControlClip2	57.72
M, 1080p	MissionControlClip3	54.40
A, 720p	sc_robot	64.84
CC, 1080p	Kimono1	27.93
Average		**56.92**

ACKNOWLEDGMENT

"This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and future Planning(NRF-2015R1A2A2A01006004)"

REFERENCES

[1] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012

[2] J. Ohm, G. J. Sullivan, H. Schwarz, T. Tan, and T. Wiegand " Comparison of the coding efficiency of video coding standards - including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669-1684, Dec. 2012.

[3] Advanced Video Coding for Generic Audiovisual Services, ITU-T and ISO/IEC JTC1, document ITU-T Rec. H.264 and ISO/IEC 14496-10, May 2003.

[4] H. Yu, K. McCann, R. Cohen, and P. Amon, Requirements for an Extension of HEVC for Coding of Screen Content, ISO/IEC JTC 1/SC 29/WG 11, document MPEG2014/N14174, San Jose, CA, USA, Jan. 2014.

[5] R. Joshi. et. al., "HEVC Screen Content Coding Draft Text 6," 23rd JCT-VC meeting, San Diego, U.S. document JCTVC-W1005, Feb. 2016.

[6] R. Joshi, J. Xu, R. Cohen, S. Liu, and Y. Ye, "Screen content coding test model 6 (SCM 6)," 22nd JCT-VC meeting, Geneva, Switzerland, document JCTVC-V1014, Oct. 2015.

[7] Z. Ma, X. Feng, and M. Xu, "Advanced screen content coding using color table and index map," *Image Processing, IEEE Transactions on*, vol. 23, no. 10, pp. 4399–4412, 2014.

 ISBN: 978-1-61208-577-7

[8] P. Lai, S. Liu, and S. Lei, “Ahg6: On adaptive color transform (act) in scm2.0,” document JCTVC-S0100, JCT-VC, Strasbourg, France, Oct. 2014.

[9] B. Li, J. Xu, G. J. Sullivan, Y. Zhou, and B. Lin, “Adaptive motion vector resolution for screen content,” document JCTVC-S0085, JCTVC, Strasbourg, France, Oct. 2014.

[10] C. Pang, J. Sole, Y. Chen, V. Seregin, and M. Karczewicz, “Intra block copy for hevc screen content coding,” *in Data Compression Conference (DCC)*, 2015, 2015, p. 465.

 ISBN: 978-1-61208-577-7

A Study on 3D Surface Graph Representations

Long H. Nguyen, Abdullah Karim
Texas Tech University, Lubbock, Texas, USA
long.nguyen@ttu.edu, abdullah.karim@ttu.edu

***Abstract*—Surface graphs have been used in many application domains to represent three-dimensional (3D) data. Another approach to representing 3D data is making projections onto two-dimensional (2D) graphs. This approach will result in multiple displays, which is time-consuming in switching between different screens for a different perspective. In this work, we study the performance of 3D version of popular 2D visualization techniques for time series: horizon graph, small multiple, and simple line graph. We explore discrimination tasks with respect to each visualization technique that requires simultaneous representations. We demonstrate our study by visualizing saturated thickness of the Ogallala aquifer - the Southern High Plains Aquifer of Texas in multiple years. For the evaluation, we design comparison and discrimination tasks and automatically record result performed by a group of students at a university. Our results show that 3D small multiples perform well with stable accuracy over numbers of occurrences. On the other hand, shared-space visualization within a single 3D coordinate system is more efficient with small number of simultaneous graphs. 3D horizon graph loses its competence in the 3D coordinate system with the lowest accuracy comparing to other techniques. Our demonstration of 3D spatial-temporal is also presented on the Southern High Plains Aquifer of Texas from 2010 to 2016.**

Keywords–Scientific visualization; time series visualizations; 3D horizon graph; space reduction; underground water aquifers.

I. INTRODUCTION

Time series charts are graphical representations of quantitative values changing over time [1]. Line chart is the simplest way to visualize time series where we have one axis represents for a time period and the other axis represents for its quantitative values. Since its invention by William Playfair (1875-1923) [2], a lot of research has been conducted and has invented different variants of time series with better visualization in certain ways. One approach that researchers try to target is the representation of multiple time series where line charts visualization become difficult because of the limited vertical space, as well as its high visual clutter. Two other popular techniques are small multiples and horizon graphs that we will discuss later in this paper.

Small multiples represent lines in its own space to reduce visual clutter. Because space is allocated per line, it requires to compact itself in order to fit inside the vertical screen resolution. This causes difficulty in value comparing between points on the same line due to the scale reduction. Later on, horizon graph was invented. In horizon graph, the gap between the line and baseline is chunked into multiple layers, and the higher layer is overlaid on top of lower ones. The color is also used to fill the gap between baseline and the area formed by the line chart and the boundaries of its layer. The horizon graph is proven as space reduction while gaining identity and information extraction solution for multiple simultaneous time series representations in 2D visualization [3]. This solves the problem of visual clutter while still representing multiple series.

Horizon graph which is represented in 2D space has grown its usage since its first invention. In recent years, people are increasing to higher demand with 3D or even multi-dimensional data expecting multi-dimensional space or even merging to virtual space to see the representation. Together with this trend, 3D visualization techniques also evolve to solve more complicated information visualization demand from multi-dimensional data. With a standard surface graph, users can deal with few simultaneous series only.

In this work, we study 3D version of split space techniques (small multiples and horizon graphs) to accommodate space and data comparison problems in the world of 3D visualization. We address the lack of perception capabilities in representing 3D data by evaluating various tasks requiring multiple concurrent displays through monitored laboratory experiments. Our main motivation for this research is to provide guidelines for graphical designers who are in need of methods to represent 3D spatial-temporal graphs for their applications. In addition, we stress advantages and disadvantages of each graph type for certain use cases. In the demo, we represent spatial-temporal data visualization of the saturated thickness of the Ogallala aquifer. In summary, our main contributions are:

- We present 3D version of horizon graph and small multiple that can address the space and perception capability in presenting multiple concurrent series.
- We provide a guideline for 3D designers in visualizing surface graph data.
- We demonstrate our sample visualizations in the Southern High Plains Aquifer of Texas and conduct a user study for each visualization technique.

We understand that providing more interactions will bring more capabilities to users in catching information from the graph. However, to evaluate the visual effect itself, we will just provide basic interaction such as zooming, and rotating graph meshes in the 3D coordinate system. Other alternative methods such as hierarchical aggregation [4], temporal queries [5], temporal clustering [6] or further interactions are not considered in the scope of this paper.

Even though our demonstration is the spatial-temporal data of the Ogallala aquifer, we believe that the technique and its analysis result can be applied to any multi-dimensional data visualization representing in 3D coordinate system.

The rest of this paper is organized as follows: We discuss the related work in the next section. Then we provide an overview of our research questions, our evaluation criteria and discuss some current 3D visualization techniques in representing surface graphs. In Section V, we discuss our study in detail with hypothesis, equipment, tasks and study design. We analyze and discuss our result in Section VI. Finally, we conclude our paper with future plans.

 ISBN: 978-1-61208-577-7

II. Related Work

In this section, we do not intend to survey all visualization and perception techniques for geospatial-temporal representations [7]. Instead, we will discuss some related research on graphical perception and some spatial-temporal visualization techniques in two or 3D coordinate systems.

A. Graphical perception

Graphical representation of statistical data evolved for years even before computers were invented. The initiative work presented by Eells et al. at [8] starting toward a higher demand which different graphical representations are compared to obtain a greater visual effect. In the next year, Croxton et al. [9] compared bar charts to circle diagrams for the accuracy of judgment. He then continued to question why most users of statistical charts found linear comparison represented by charts has more accuracy than other area charts comparison. He discussed the related capability of bars, squares, circles, and cubes by some simple comparisons at [10]. Peterson et al. at [11] tried to figure out "How accurately are different kinds of graphs read?" by reading values from different graphical representations of statistical data.

Cleveland et al. at [12] set scientific foundation for graphical analysis and data representation under the name *graphical perception* and defined it as the visual decoding of information encoded on graphs. Simkin et al. at [13] confirmed by evidence that people have generic expectation about types of information will be the major messages in some types of graph. A comparison is conducted and revealed that the most accurate of judgment is in bar charts then divided bar charts and least accurate is the pie charts. However, the representation involves only two charts simultaneously.

Lohse et al. at [14] presented a cognitive model to understand perceptual and cognitive processes people use to decode information from a graph. They conducted an empirical study in predicting reaction time, level of difficulty to acquire information between computer simulated graphical perception and actual performance of users. After two years, Gillan et al. [15] discussed a componential model of human interaction with the graphical display. From human-computer interaction point of view, the model explains human interaction with graphs. Later on, he brought this model to study with various graphical representations, such as line graphs in linear modeling, scatter plots, stacked bars, and pie charts.

Huang et al. [16] argued that time and error measures are limited in providing essential knowledge that is useful for graphical design. They conducted three user studies to evaluate and demonstrate the usefulness of cognitive perspective that goes beyond merely measuring time and error. Spence et al. [17] used a series of experiments to investigate the perception of percentages in bar charts, pie charts, and tables. And very recently, Heer et al. [18] conducted two controlled experiments to measure the effect of chart size and layering and evaluate their effect in value comparison tasks.

Barfield et al. at [19] discussed the relationship between 2D or 3D graphs displayed on paper or computer and the problem-solving performance of experienced and novice manager with respect to each representation by measuring solution times, confidence in answers and effectiveness of solutions.

Each of these studies are focus on how a visual encoding and representing affects the accuracy and/or response time of discriminating values of the underlying data. Nevertheless, most of these researches discussed those visual factors with the 2D visualization techniques. There are still remaining questions on how well these techniques perform when dealing with multi-dimensional data and representing as 3D charts.

B. Spatial-temporal data visualization

One of the earliest demonstrations related to spatial temporal data is Napoleon's march towards Moscow. Its visual encoding uses color, size, text annotation, and position to represent different states of Napoleon's army in 1812 [20] in one snapshot. Nowadays, data visualization tools become more important when the demand in data dimension increased. David et al. [21] presented abstract interface using recent web based JavaScript libraries to visualize large complex sets of spatial temporal data over the web.

Dübel et al. [22] shown that 2D and 3D visualization techniques exhibit different advantages and disadvantages in human perception. Aigner et al. [23] discussed analytical view on factors, and providing samples to visualize time oriented data. Andrienko et al. [24] introduced the use of interactive visualization tools to address problems of spatial temporal data.

Tominski et al. [25] brought 3D icons into a map display for representing spatial-temporal data. There are also event-based methods being integrated for reducing the amount of information to be represented. The approach relies on two popular concepts: 3D information visualization and information hiding. The technique is inherited by three-step process for information visualization: overview first, zoom and filter, and then details-on-demand which were presented by Keim et al. [26].

In supporting with the third dimension, Carneiro [27] presented a case study of Geneva with 3D Geneva project. It aims to evaluate if users would potentially be interested to integrate the third dimension in visualizing the available geographical data. Koussa et al. [28] proposed web based solution for 3D geographical information system using multilevel spatial database structure and layer management technique to visualize and analyze the data.

Treinish [29] discussed human problem solving and decision making performance varies enormously (100:1) with different presentations of high-resolution incorporated predictive weather data. More recent work was done by Kehrer et al. [30] who summarized existing methods for visualization and interactive visual analysis of multifaceted scientific data and proposed a categorization of approaches.

III. Definitions, Data Set and Methodology

In this section, we discuss some common definitions that are used throughout the paper. We also briefly introduce our data set and methodology in this research.

A. Definitions

Saturated thickness is the vertical thickness of a hydrogeologically defined aquifer in which the pore spaces of the rock forming the aquifer are filled with water [31].

 ISBN: 978-1-61208-577-7

Graph type is the type of graph used in representing information. In this research, graph type is either surface graphs, small multiples or horizon graphs. If we do not mention whether these types are 3D or 2D, it is implicitly understood that the graphs are in 3D coordinate system.

Study year is the year that the study data belongs to. When discussing its graphical representation, we mean that it is a surface graph representing the data of that year.

B. Data Set

We use saturated thickness data from 2010 to 2016 to create visualizations in the study. We choose this data set because it has both spatial, temporal dimensions which fulfill our research objective in this work. Besides, these data sets are part of our research project in visualizing the aquifer.

C. Methodology

Our research goal is to effectively visualize the saturated thickness of the Ogallala aquifer in a 3D coordinate system that requires little space but also maintains the visual richness necessary for comparison tasks. To do so, we first designed a model in which the geographical locations encompassing the Ogallala were used as the base ground. Because this model considered only the thickness of the Ogallala at a given location, these locations described a flat plane. The surface of the model represented the saturated thickness of the Ogallala, producing an image resembling a terrain and so called a surface graph. We represent the surface graph via different techniques (simple surface graph, small multiple and horizon graph) and conduct a user study to evaluate its effectiveness based on user perception.

IV. Evaluation Criteria and Visualization Techniques

In this section, we discuss evaluation criteria and visualization techniques individually to justify each case.

A. Evaluation Criteria

We reuse some evaluation criteria described in [32] to evaluate each surface representation. Table I summarizes attributes for the three surface techniques surveyed in this paper. We discuss these techniques in the following section.

- **Space management**: This criterion describes how space is utilized in a graphical representation. In other words, it is whether space is shared or split into multiple occurrences. Shared space is represented in the same coordinate system. So, all series are overlaid; hence, it is easier for comparison. The height of the space required is proportional to the biggest quantitative value of the data in all study years. However, if there are more study years, it may introduce clutter which makes it hard for users to identify the graphs and compare one to another. On the other hand, in split space visualization, whether there are less or more occurrences, each graph has its own space to be presented. It means that this technique requires the space to be compressed to fit all the graphs. Hence the perception of information may be reduced.
- **Space per study year**: This is the vertical and horizontal (a rectangular cube) amount of display space allocated to each graph. This characteristic is not a key factor, but it is important as any visualization should fit in a monitor screen size. We aim to use as little space as possible without negative consequences to its visualization effect.
- **Identity**: This criterion tells how easy it is to distinguish between occurrences. Obviously, it is more difficult to identify graphs in shared space visualization. It is required to use graphical methods such as colors and styles to convey the graph identity. The split space techniques preserve the identity of the graphs.
- **Visual clutter**: This factor refers to clutter associated with representation techniques for a large number of graphs.

In addition, we add a new criteria to evaluate the effectiveness of the representation technique, called *information identification*.

- **Information identification**: One of the key factors that affects user preference for one graph type over others is its information identification property. One graph stands out from other graphs because it conveys information clearly and makes it easy to find information within the graph. To measure this factor, we can use its accuracy test and the time it takes to complete related tasks.

TABLE I. PROPERTIES OF 3D VISUALIZATION TECHNIQUES SURVEYED IN THIS PAPER.

Visualization Technique	Space Management	Space per Study Year	Visual Clutter	Information Identification
Simple Surface	Shared	S + h	Medium	Medium
Small multiple	Split	S/N + h	Low	Medium
3D Horizon	Split	S/(N*2*B) + h	Low	Low

In Table I, **S** is the available space, **N** is number of study years, **B** is number of bands and **h** is the minimum height required to view the entire surface (to view the depth level of the surface).

B. Simple Surface Graph

Simple surface graph is a common graph that we represent in 3D space. Each axis represents one dimension of the data. In our demonstration, the ground plane (Oxy) is the longitude and latitude, and the vertical (Z) axis is the saturated thickness. With this type of graph, we can intuitively see that the space required to represent is shared between graphs and equal to the rectangular cube surrounding the vertical height and horizontal width of the graphs.

Adding more graph surface is achieved by presenting it in the same coordinate system and with different surface colors or styles to identify itself. To make the 3D effect more efficient, we can use the same color with different bands or a line grid along the surface to improve differentiation of height levels in the graph. In this paper, when we overlaid multiple study years, we received informal feedback that multiple colors make confused users, so it is unfair in comparison with other techniques. Therefore, we decided to keep one color for each study year.

Figure 1 shows an example with four study years of saturated thickness of the Ogallala. Each study year is represented

 ISBN: 978-1-61208-577-7

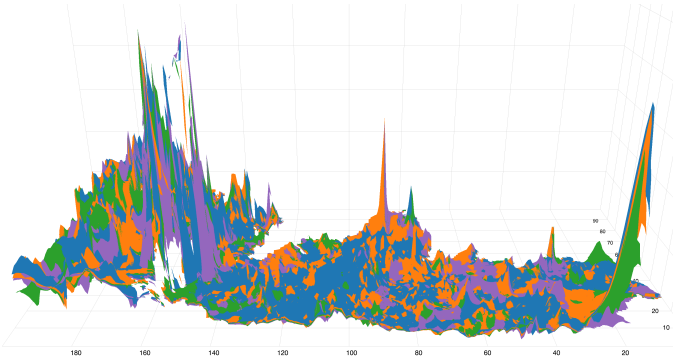

Figure 1. Saturated thickness surface graphs of the Ogallala in 2010 (blue), 2012 (orange), 2014 (green) and 2016 (purple) in the shared space visualization technique.

in a separate color. As shown in Table I, these graphs are shared space so that we can easily compare saturated thickness over years. Distinguishing identity is already challenging because the surfaces are overlaid and hide each other. The mix of the colors with the complexity of each graph introduces clutter as well.

C. Small Multiple

Small multiple for surface graph visualization means that we split the space into sub spaces separated by a plane. Each graph lies inside the allocated space and has its own coordinate system. However, these coordinate systems have the same scale to maintain comparison capability. In other words, with this type of graph, we no longer add graphs into the same graph space hence it is critical to have the same axes scaling across all allocated spaces to allow comparison between graphs. The more we compact the graph, the shorter and smaller amount of space we require to represent them. However, to a certain limit, we can't compress the space more because it will be harder to tell whether it is a 3D mesh or just a thick plane.

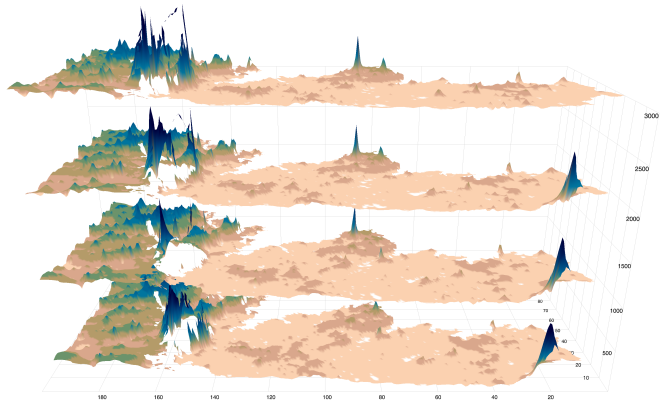

Figure 2. Saturated thickness surface graphs of the Ogallala in 2010, 2012, 2014 and 2016 (from bottom to top) with small multiple representation.

In Figure 2, we demonstrate small multiples of three study years of saturated thickness of the Ogallala aquifer. We rescale the height of the saturated thickness in accordance with the scale of the coordinate system in each allocated space. This made the saturated thickness look shorter compared to its original simple surface graph. Obviously, the space to represent each graph is now horizontally and vertically reduced to **(S / N) + h** as demonstrated in Table I. In addition, we gain the identity property in this case since each year representation is separated. The saturated thickness value can be measured by its height or color represented in the graph. This feature makes 3D small multiple different from its 2D version that the line height is only measured by its value in the Y axis.

D. 3D Horizon Graphs

The horizon graph was originally represented as two-tone pseudo-coloring [33]. Latter on, it was developed and introduced under the name "Horizon graph" by the company Panopticon [3]. The graph is chunked into multiple layers. The upper layer is overlaid on top of a lower layer. Each layer is encoded with a different color to convey its height information. Its construction steps are represented in [3].

We reuse the concept of horizon 2D in 3D horizon graph visualization that upper layer of the graph is moved down to the base ground plan. Its color band is still the same as other visualizations. The allocated space per graph is reduced to **(S / (N * 2 * B)) + h** as described in Table I. This makes the horizon graph use the least space compared to other techniques we have discussed. Figure 3 provides an example of 3D horizon graphs for four study years.

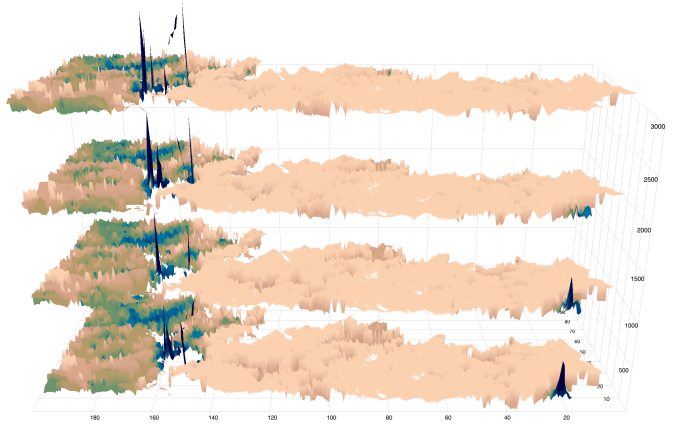

Figure 3. The 3D horizon graphs of the Saturated thickness of the Ogallala in 2010, 2012, 2014 and 2016 (from bottom to top).

V. User Study

Our intention of this research is to introduce a 3D version of horizon graph and small multiple, and explore user performance on these techniques under different space and cardinality constraints. Does the 3D horizon graph perform better over other techniques? How does each type of graph benefit users in certain conditions?

To investigate these research questions, we designed a quantitative user study approach to measure correctness performance and time it takes for different combinations of visualization techniques and number of study years. In particular, we present interfaces of surface graphs, small multiples and 3D horizon graphs. With each graph type, we present two, three and four study years for comparison and discrimination tasks. The measurement matrices are correctness of each task's answer and the time requires to complete each trial.

A. Hypothesis

It is intuitive that shared space surface graph and split space graph techniques have different strengths and weaknesses for

 ISBN: 978-1-61208-577-7

tasks with different visual spans. Also the number of study years would strongly affect to user legibility.

- **H1** *Simple surface graph will outperform other techniques on small number of surfaces*(for example, with 2 surfaces sharing the same coordinate). We predict that, the simple graph will outperform with few study years when the visual clutter is small and has less negative effects. When the number of study years increases, the visual clutter becomes more problematic and deducts graphical perception.
- **H2** *Small vertical resolution will reduce small multiple performance*. We want to confirm this premise that the display space will have a strong impact to user performance.
- **H3** *Small multiples out perform other techniques when there are more occurrences*. Besides the height attribute, 3D small multiples also use color scale to represent the height property in order to improve its legibility in a compact space. Horizon graph technique does not perform as well as it does in the 2D visualization comparing to the small multiple technique.

B. Visualization Tasks

There are many tasks related to assessing graph representations. However, our hypotheses are based on the premise that each graph visualization technique has different strengths and weaknesses for different tasks. So through in-depth discussions with water resource experts, taxonomists, and ontology researchers, we identified two primary tasks important for visual identification of saturated thickness of the Ogallala:

- **Maximum**: a simple exact location comparison across all study years [34].
- **Discrimination**: a dispersed location comparison between study years [13].

1) Maximum (Exact location comparison): We require participants in this task to find in a study year with the highest value at a specific location. So within a single graph type, there are several study years. We defined a location A that has the same longitude and latitude in all study years and ask participants for the year with highest saturated thickness.

2) Discrimination (Dispersed location comparison): This task expects users to determine which study year has highest saturated thickness value at a location specific to each study year between graph types. We randomly select a location in each study year and ask participants to find the highest saturated thickness across all study years.

C. Environment and equipments

The experiments were conducted in a monitored laboratory which each user gets a short training before doing the tasks. Then we explain the tasks to users and ask him/her for a trial to get familiar with the test tool before doing actual tests. All experiments were conducted on a standard iMAC desktop computer 27 inches equipped with a keyboard and a mouse. The screen is set to 5120 x 2880 resolution. The test application was maximized on the screen. Participants only used the mouse during the experiments.

D. Participants

We recruited 10 subjects (5 males, 5 females), aged between 22 to 25. Participants were all volunteers and had normal or corrected normal vision with no color blindness. We also conduct screening to make sure students capable of using computer and have some graph experience. For these reasons, sophomore students or above who had taken courses that have graphical representation get selected.

E. Trial condition

Our one shot trial test has a monitored number of simultaneous study years displaying on the screen. There is only a single graph type representing study years on each screen to avoid confusion to participants. With simple surface graph, we represent each surface with its own color. The other graph types have color scale to represent the height of saturated thickness. The darker the color is the higher value of saturated thickness we observe.

All trials are run with full window screen with only zooming and rotating interaction. These factors are used to allow user to rotate and understand other faces of the graphs or look into details on zooming experience. To answer any test questions, there will be a form with radio buttons for selection of correct answer(s) and a confirmation before the answer data transferred to our recording system.

F. Procedure

First we train participants with explanation of the surface graphs, small multiples and 3D horizon graphs. We also introduce simple interaction with graphs such as zooming and rotating to view different faces of the graph.

After the training phase, participants are required to do practice test to make sure the test application is understood, and become familiar with the graphs. They are also required to explain concepts of each graph type to confirm theoretical and practical understanding. In addition, we introduce the concept of saturated thickness, and how it is described in the graphs. Participants sit in front of the monitor at comfortable distance (around 50cm) to them. It is possible to ask questions during the training phase but it is not allowed for the real trials.

Participants are required to finish all trials for a particular graph type before moving to other ones. They are instructed to complete the trials as quickly as possible. To avoid clicking mistakes, there is always a confirmation button for selection before moving to the next question.

G. Study Design

The following factors are included in our study:

- **Visualization type (V)**: Simple surface graph, small multiple and 3D horizon.
- **Number of simultaneous displays (N)**: 2, 3 and 4 years.
- **Task (T)**: Maximum and Discrimination.

From these factors, we have *V x N x S* or *3 X 3 X 2 = 18* combinations for trial conditions. Each condition is repeated twice so we have a total of **36** trials per participant. The tasks are ordered from simple to complex for user to be prepared at the end. The order of graph types and order of number of study years are random.

 ISBN: 978-1-61208-577-7

There are **10** participants and each has **36** trials. Hence, the study system collects for a total of **10 x 36 = 360** trials for entire experiment.

VI. Results And Explanation

We consider accuracy as the key factor for evaluation. Since it does not matter if participants perform extremely fast within a given visualization, but in the end of the test session, the answer is incorrect. We will evaluate overall task completion accuracy and time to see the impact of number of study years. Then, we will look into comparison about accuracy and completion time of two tasks to see how graph type affects perception.

A. Average completion accuracy and time

From Figure 4, we can confirm our hypothesis **H1** again: the simple surface graph is more accurate with small number of occurrences (two occurrences) and it dramatically drops by number of study years from 90% to 68% at four occurrences. The 3D horizon graph always has lowest accuracy rate. Its performance drops quickly from 65% at two occurrences to 42% at four occurrences. On the other hand, the small multiple graphs slightly decrease percentage of accuracy. The change of percentage is not significant so we can tell that the small multiple technique stably maintains visual effect compared to other techniques when number of study years increases. In particular, the small multiple technique starts to get more accurate from 3 occurrences compared to normal surface graph (hypothesis **H3**). However, the accuracy is deducted with all graph types (hypothesis **H2**).

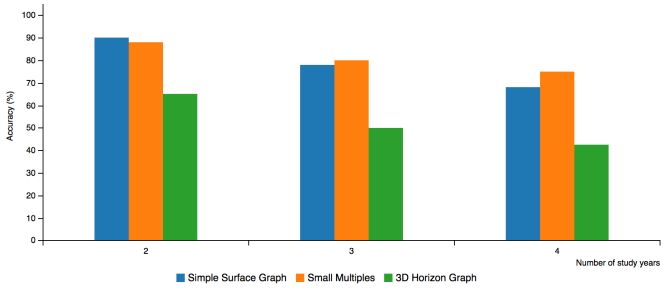

Figure 4. Percentage accuracy (in %) of tasks by number of study years

Below, we plot the average completion time of all tasks from our results. Figure 5 indicates that when number of study years increases, it requires more processing time in perception in order to do the task. It doubles the perception time in case of simple surface graph, from 09 seconds with two study years until around 19 seconds to complete the same task with four years representation. In contrast, the small multiple and 3D horizon graph gradually increase its completion time that it can be explained by the complexity of the representation.

B. Breakdown of completion accuracy by tasks

To understand how graph types make a different in performing tasks (simple and complex tasks), we visualize the relationship between accuracy difference between the two tasks (**T1 - T2**) and number of study years as line graph. Each line stands for a graph type and its height is the accuracy difference. If the performance gap keeps increasing by study year, it means

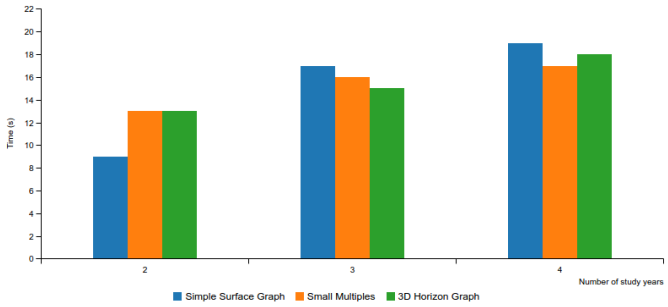

Figure 5. Average completion time of tasks by number of study years

that the visualization technique has poor representation for number of occurrences.

From Figure 6, we found that task **T1** is always more accurate than task **T2** across all graph types (Because the points are all positive). The 3D horizon graph always has a higher gap and increases quickly from 10% with two occurrences to 35% differences with four occurrences. The small multiple slightly increases and keeps stable between three and four occurrences.

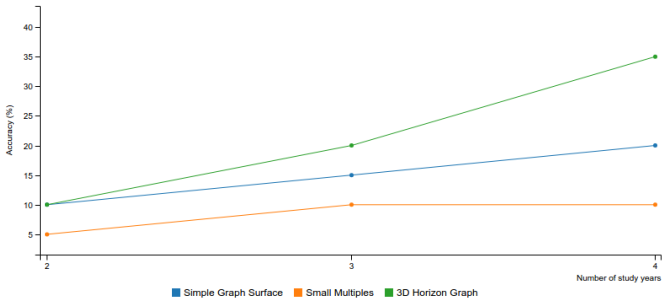

Figure 6. Accuracy difference between tasks (**T1 - T2**) by study years

From these analyses, we can see that graph type does affect user perception. It helps users to capture information in different ways so that the accuracy and time does vary with each representation. Depending on the need, 3D graphic designers can decide the graph type to visualize the information.

VII. Conclusion And Future Work

In this paper, we present the results of a user study approach in understanding perception regarding surface graph representations in 3D space. The results show that a few number of simple surface graphs is best for the maximum task however it shows huge clutter or overlapping resulting in information hiding with more occurrences in the graphical space. Moreover, the space that is required to visualize the graph is proportional to the saturated thickness of the graphs. The small multiple technique outperforms for larger numbers of study years. Even though the 3D horizon graph has the most compact space, its accuracy is still bottleneck.

It is obvious that the 3D representation looks more elegant compare to the 2D representation. Therefore, we are excited about our future work, which is to improve the performance of the 3D horizon graph by providing more interaction such as a slider to look into any slice of the graph mesh. Each slice represents the projection of the graph cut onto a plane. We will then investigate if this additional view will bring better information capturing for user. In addition, we will investigate

 ISBN: 978-1-61208-577-7

more formal validation methods with support of statistical testing to ensure the correctness of our methodology.

REFERENCES

[1] "Time series," https://en.wikipedia.org/wiki/Time_series [Accessed date: June 17, 2017].

[2] "William playfair," https://en.wikipedia.org/wiki/William_Playfair [Accessed date: June 17, 2017].

[3] R. Hannes, "The development of the horizon graph," 2008.

[4] N. Elmqvist and J.-D. Fekete, "Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines," IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 3, 2010, pp. 439–454.

[5] H. Hochheiser and B. Shneiderman, "Dynamic query tools for time series data sets: timebox widgets for interactive exploration," Information Visualization, vol. 3, no. 1, 2004, pp. 1–18.

[6] L. Kaufman and P. J. Rousseeuw, Finding groups in data: an introduction to cluster analysis. John Wiley & Sons, 2009, vol. 344.

[7] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and C. Sheelagh, "A review of temporal data visualizations based on space-time cube operations," in Eurographics Conference on Visualization. Eurographics, 2014.

[8] W. C. Eells, "The relative merits of circles and bars for representing component parts," Journal of the American Statistical Association, vol. 21, no. 154, 1926, pp. 119–132.

[9] F. E. Croxton and R. E. Stryker, "Bar charts versus circle diagrams," Journal of the American Statistical Association, vol. 22, no. 160, 1927, pp. 473–482.

[10] F. E. Croxton and H. Stein, "Graphic comparisons by bars, squares, circles, and cubes," Journal of the American Statistical Association, vol. 27, no. 177, 1932, pp. 54–60.

[11] L. V. Peterson and W. Schramm, "How accurately are different kinds of graphs read?" Audiovisual communication review, vol. 2, no. 3, 1954, pp. 178–189.

[12] W. S. Cleveland and R. McGill, "Graphical perception: Theory, experimentation, and application to the development of graphical methods," Journal of the American statistical association, vol. 79, no. 387, 1984, pp. 531–554.

[13] D. Simkin and R. Hastie, "An information-processing analysis of graph perception," Journal of the American Statistical Association, vol. 82, no. 398, 1987, pp. 454–465.

[14] J. Lohse, "A cognitive model for the perception and understanding of graphs," in Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 1991, pp. 137–144.

[15] D. J. Gillan, "A componential model of human interaction with graphical display," ACM SIGCHI Bulletin, vol. 25, no. 3, 1993, pp. 64–66.

[16] W. Huang, P. Eades, and S.-H. Hong, "Beyond time and error: a cognitive approach to the evaluation of graph drawings," in Proceedings of the 2008 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization. ACM, 2008, p. 3.

[17] I. Spence and S. Lewandowsky, "Displaying proportions and percentages," Applied Cognitive Psychology, vol. 5, no. 1, 1991, pp. 61–77.

[18] J. Heer, N. Kong, and M. Agrawala, "Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2009, pp. 1303–1312.

[19] W. Barfield and R. Robless, "The effects of two-or three-dimensional graphics on the problem-solving performance of experienced and novice decision makers," Behaviour & Information Technology, vol. 8, no. 5, 1989, pp. 369–385.

[20] E. R. Tufte, The Visual Display of Quantitative Information. Cheshire, CT, USA: Graphics Press, 1986.

[21] A. David and C. J. Tauro, "Web 3d data visualization of spatio temporal data using data driven document (d3js)," International Journal of Computer Applications, vol. 111, no. 4, 2015.

[22] S. Dübel, M. Röhlig, H. Schumann, and M. Trapp, "2d and 3d presentation of spatial data: a systematic review," in 3DVis (3DVis), 2014 IEEE VIS International Workshop on. IEEE, 2014, pp. 11–18.

[23] W. Aigner, S. Miksch, H. Schumann, and C. Tominski, Visualization of time-oriented data. Springer Science & Business Media, 2011.

[24] N. Andrienko and G. Andrienko, "Interactive visual tools to explore spatio-temporal variation," in Proceedings of the working conference on Advanced visual interfaces. ACM, 2004, pp. 417–420.

[25] C. Tominski, P. Schulze-Wollgast, and H. Schumann, "3d information visualization for time dependent data on maps," in Information Visualisation, 2005. Proceedings. Ninth International Conference on. IEEE, 2005, pp. 175–181.

[26] D. A. Keim, C. Panse, and M. Sips, "Information visualization : Scope, techniques and opportunities for geovisualization," in Exploring Geovisualization, J. Dykes, Ed. Oxford: Elsevier, 2004, pp. 1–17.

[27] C. Carneiro, "Communication and visualization of 3-d urban spatial data according to user requirements: case study of geneva," in Proceedings of the XXI ISPRS Congress, vol. 37, 2008, pp. 631–636.

[28] C. Koussa and M. Koehl, "Proposal of 3d gis for spatial data visualization and analysis over internet."

[29] L. A. Treinish, "Visual data fusion for applications of high-resolution numerical weather prediction," in Proceedings of the conference on Visualization'00. IEEE Computer Society Press, 2000, pp. 477–480.

[30] J. Kehrer and H. Hauser, "Visualization and visual analysis of multifaceted scientific data: A survey," IEEE transactions on visualization and computer graphics, vol. 19, no. 3, 2013, pp. 495–513.

[31] R. W. B. J. A. Schloss, "Current saturated thickness," http://www.kgs.ku.edu/HighPlains/atlas/atcst.htm [Accessed date: June 17, 2017], last revision 2000.

[32] W. Javed, B. McDonnel, and N. Elmqvist, "Graphical perception of multiple time series," IEEE transactions on visualization and computer graphics, vol. 16, no. 6, 2010, pp. 927–934.

[33] Saito et al., "Two-tone pseudo coloring: Compact visualization for one-dimensional data," in Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on. IEEE, 2005, pp. 173–180.

[34] H. Lam, T. Munzner, and R. Kincaid, "Overview use in multiple visual information resolution interfaces," IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 6, 2007, pp. 1278–1285.

 ISBN: 978-1-61208-577-7