THE CHALLENGE OF ANALYZING GEOVISUALIZATION TOOL USE: Taking a visual approach

Dan Haug, Alan M. MacEachren, and Frank Hardisty
GeoVISTA Center (http//www.geovista.psu.edu/), Department of Geography,
302 Walker Penn State University, USA;
Fax: 914-863-7943; Email: alan@geog.psu.edu

Abstract: *Over the past decade, there has been a dramatic change in map forms and map-based analysis. In the past, maps have been designed by cartographers to be studied and interpreted by users. Today there are widely available mapping environments that allow users to generate their own maps and dynamically manipulate parameters of those maps. Although research advances in dynamic mapping are being rapidly incorporated in commercial mapping and GIS products, little is understood about the use of interactive maps or how that use influences tasks such as knowledge construction or decision-making. One of the most serious impediments to developing this understanding is that we lack methods and tools to study dynamic map use. This paper presents a framework for the development of a software environment that will facilitate cognitive and usability studies directed to understanding use and improving usefulness of interactive geovisualization environments.*

1 Introduction

The paper begins with a brief outline of techniques used for evaluating software and software use, focusing on the goals of evaluation, methods that meet those goals, and the types of data that are required to execute these methods. The paper then discusses an approach to collecting and organizing these data, focusing on the problem of referencing multiple data streams of different types to one another so that the analyst can explore patterns between those data streams. Finally, we discuss a set of software tools currently under development for the purpose of conducting these types of analyses, and extending them with new functionality designed specifically to meet the needs of interactive visual environments.

2 Evaluating Software and Software Use

There are a broad number of techniques for evaluating software and the use of software. In this paper we focus on empirical techniques in which we collect data by directly observing users interacting with the software, and then use those data to evaluate the software. Such evaluation can be designed to meet different goals. After a brief discussion of some different types of goals, we address types of data that can be collected, and methods of analyzing those data, both of which may vary depending on the goals of the analysis.

2.1 Goals for evaluation

Slocum et. al (2000) distinguish between studies that have the goal of improving software design (within the framework of how useable it is), and studies that are designed to extend theoretical knowledge of human cognition, and how software can be designed to better match existing cognitive structures. They refer to the first of these goals as "usability" assessment, and the second as "cognitive" testing.

2.1.1 Usability Assessment

Usability assessment is best considered to be an iterative process through which usability is assessed and improved at multiple stages during the software design process, from early system design (at stages before any code is being implemented) through deployment (including stages where the software has become a legacy product) (see Nielsen 1993 and Schneiderman 1998). Assessment of aspects of system usability can be conducted in a number of ways. These include: automated testing using other software products, evaluation by software development experts, evaluation by usability experts, evaluation by the users themselves, or a combination of these. Assessment can also be broken down into types depending on the goals for the evaluative process. This is an important distinction for our research. Buttenfield (1999), in discussing usability of digital geospatial libraries, points out that the goals of assessing usability of software can be either formative or summative (a standard distinction in the usability engineering literature – see Gabbard et. al 1999 for related ideas in the context of virtual environment design). Formative evaluation has the goal of modifying and improving the software. Summative evaluation has the goal of determining whether or not the system (or components of the system) meets its objectives, or comparing one system to another. Our work is intended to support a range of formative and summative assessment methods. Thus far, we have devoted particular attention to developing tools that can be applied to specific summative assessment tasks (e.g., comparing the usability and usefulness of two alternative tools for depicting multivariate data relationships).

2.1.2 Cognitive Testing

The goal of cognitive testing in the context of geovisualization and related software is to improve our understanding of cognitive processes that may be employed when using specific software tools and to devise tools that provide better support for those processes (see Mark et. al 1999 for a review of cognitive research in cartography, including interactive maps). This can be especially important in the context of software for exploration and hypothesis formation. Knowledge construction and decision-making are cognitive processes (key processes in hypothesis generation), which cannot be directly observed. Because of this, methods for investigating these processes involve inference from more observable phenomena. In most cases this means tracking what study participants say and do while they are interacting with dynamic maps, recording their stated conclusions that are based on those observed interactions, and then developing a model that explains how the process worked (see Chen and Czerwinski 2000 for specific examples). In order to do this effectively, we need to be able to know what the participant is *doing, saying,* and *seeing* at any given moment. In addition, we need to be able to compare patterns in these data between different participants.

2.2 Methods for Formative Evaluation and Cognitive Testing

While it is important to take into account the goals of a specific software evaluation study when choosing methods, most methods can be used to meet a broad range of goals. Formative evaluation is often conducted through methods such as interpretive field studies (Klein & Meyers 1999), cognitive walk-throughs (Buttenfield 1999), or interviews. Summative studies tend to focus on more formal methods such as protocol analysis and task-based performance measures (James and Sandersen 1991) and task analysis (Morse et. al 2000), although these methods can be—and have been—used in formative evaluation as well. However, the time that it takes to conduct studies using these methods are an impediment to their use in formative assessment, which is often rapid and iterative.

While we believe that the tools that we are developing will support a broad variety of methods of analysis, this paper mainly focuses on exploratory forms of analysis which take advantage of a combination of highly structured user interaction logs, talk-aloud protocols, and well as data tracking the display.

3 Software tools

We have begun to implement a set of software tools that take into account the types of software evaluation that can be conducted as well as the different goals for such evaluation. The remainder of this paper reports how these tools will collect and organize data and how they will support different types of analysis of those data.

We plan to focus the development of analysis tools in two complimentary areas. The first is the use of qualitative analysis of data in which we can construct links between what participants are *saying*, what participants are *doing*, what is being displayed on the monitor (as a proxy for what participants are *seeing*), and theoretical constructs that help us make sense of these data. The second is the development of a suite of interactive visual tools, based on the principles of exploratory data analysis, which support the investigation of patterns of software use.

3.1 Collecting and Structuring Data

The usability assessment-cognitive testing tools we are developing are being designed to support analysis using two primary types of data: interaction logs and talk-aloud protocols (in a number of different forms). In the future we may extend our system to support digital audio and video data, as we believe that the principle functionality developed for interaction logs and protocol data can naturally be extended to these types of data as well.

3.1.1 Chunking Data

One of the first decisions that needs to be made when collecting and analyzing data is how the data are going to be structured. Qualitative data analysis (QDA) literature has framed the problem of data structures in terms of "chunking" data, or deciding what units the data will be partitioned into for the purposes of analysis (Dey 1993). This section discusses tools for chunking both protocol data and interaction logs.

3.1.1.1 Talk-aloud protocol data

The chunking of textual data is a task that is well examined in the QDA literature (see Coffey and Atkinson 1996). These data can be seen as an essentially continuous narrative that requires someone to decide how it should be broken into units for the purposes of analysis. For instance, data that are transcribed and coded from audio or video recordings are broken into sections that seem logical to the coder. The breaks can be based on pauses in the recording, changes of topic, or changes in who is speaking. Unfortunately, making chunking decisions on the fly can slow down the coding process. One alternative to this approach (outlined by Nyerges et. al 1998) is to code continuous data in regular units of time (such as one minute). Nyerges et. al refer to this as time-based coding—as opposed to the "event-based" coding described above. It has the advantage of reducing the cognitive load on the person doing the coding, however it does not remove the problem of deciding how to chunk the data. The decision is simply made based on the goals of the study and the nature of the *entire* dataset, rather than in the specific context of one "event".

Whether an analyst chooses to use event-based or time-based chunking, there is another key problem that still may arise. In many cases the study participant may cover a number of key analytical issues that overlap in a single chunk. In these situations there may not be one "best" way to chunk the data, or break it into analytical units.

Qualitative data analysis software packages (such as NU*DIST, Nvivo, and Atlas/TI) have addressed this problem by *not* chunking the narrative itself. Instead, these products leave the narrative as a continuous text, and superimpose "codes" on variable units of the text. One section of text can be chunked into many different units of analysis for different purposes. Each chunk of data is coded by linking it to a concept that the analyst believes it relates to. There are a number of formal methods outlining this process of linking the words spoken by research participants to theoretical constructs (see Denzin and Lincoln 1998). Arguably the most rigorous of these is grounded theory. Grounded theory outlines a process by which an analyst can link chunks of data to progressively more abstract theoretical constructs through a hierarchical coding structure (Glaser and Strauss 1967 and Strauss and Corbin 1990).

One of the key strengths of grounded theory is that it provides the flexibility to explore the *meaning* of narrative data (a common feature of QDA methods) while providing a rigorous methodology. The result is a hierarchical structure linking the data directly to the conclusions of the study, thus making the analytical process more open for critical review.

3.1.1.2 Interaction log data

Interaction log data need to be chunked into units of analysis in the same way that narrative data do (Bayer et. al 1999). One major difference, however, is that interaction log data are often collected at a coarser resolution than narrative data. For instance, many interaction logs simply represent interaction in terms of units defined in the user interface. When the user employs a zoom-window tool to increase the scale of the map, this is recorded as a single interaction. However, the act of zooming could just as easily be conceived of a series of events, such as a mouse-down, followed by a series of mouse-move (mouse-drag) events, and concluding with a mouse-release. It could, surely, be broken down even further into the machine-level interactions between the different components of the computer as well.

While this is an example of a finer resolutions for capturing interaction log data, one could just as easily, and perhaps more productively, define larger chunks of interaction data. For instance, consider a zoom action that was preceded by an interaction where the user pushed the “play” button to begin an animation. The system may or may not track the steps of the animation just as it may or may not track the end of the animation (if the animation ends without user intervention). In any case, the user may conceive of the animation coupled with a following zoom as being a single task. Knowing this may be extremely important if the purpose of the study is to identify cognitive structures that participants draw upon when developing strategies for exploratory data analysis. Furthermore, unless the data are captured at the proper resolution, it may be impossible to precisely define what the user sees as a task, and thus impossible to use those tasks as an entrée into cognitive structures.

The point is that we need to collect data at a fine enough resolution to facilitate chunking at different levels. This process should directly parallel the previously discussed process of chunking narrative data. One of the advantages that interaction log data have over narrative data is that if the system designers are careful in the way they design the software that they are testing, they can automatically chunk the data at multiple resolutions. Thus the analyst has the choice of using predefined chunks of data (operations) or defining chunks of data through a manual or semi-automated analysis process. For example, individual events can be tracked by the system, but a hierarchical data structure can be employed to group a set of events into a low-level interaction (such as zooming). Furthermore, if the data structure supports it, those low-level interactions can be grouped together into conceptual interactions at a higher level – either automatically or manually. This is in contrast to chunking of narrative data, which (for the most part) takes place through an arduous manual analysis process.

In the same way an analyst can chunk the data, users could be allowed to review their sessions and chunk the data themselves. As the earlier example pointed out, sometimes users may have different ideas of what constitutes a functional unit of interaction. If this is the case, it may be useful to compare the different ways in which study participants conceive of units of interaction. In order to do this, it may be necessary to allow different users to chunk a single set of interactions in a number of different ways. It may, in fact, be quite useful to allow multiple people to independently chunk the logs into what they consider meaningful units. If certain operations are chunked in a consistent way by a number of different users, it may indicate the existence of a shared cognitive process or structure. Using this information, the analyst may be able to design a more rigorous method to formally test for such a process or structure.

This capability would allow the analyst to not only trace higher-level (possibly cognitive) operations to low-level interactions, it would allow her to map one person’s conception of operations to another’s. This may be a particularly useful capability when conducting cognitive research, where the goal is to circumscribe common cognitive structures that may effect choices in the way a user approaches specific tasks, and could be equally useful in determining whether groups of users share certain strategies for solving specific types of problems.

Of course, such analysis would benefit from the ability to link interaction log data to narrative data. This link could be achieved through a time-stamp on both the narrative data and in the interaction logs (as in MacShapa). We are designing a data structure that could be used to represent the combination of coded narrative data and coded interaction log data. It essentially consists of two temporally linked parallel data streams, each with it’s own set of hierarchical codes. It obviously could become quite complicated. This raises the problem of how one goes about making sense of such a dataset. The issue is complicated even further when we take into account the third aspect of our analysis: what the user is seeing.

3.1.2 Tracking the display

Adding another dimension to this complex collection of information is the fact that, in evaluating visualization software, what was actually being displayed on the monitor at any given time during data collection is of utmost importance. After all, there may be a dissonance between what the user is *saying* and what is on the display. Such dissonances (if found to be consistent) can lead to insights about both perceptual and cognitive factors in human computer interaction.

There are few documented methods for dealing with the visual aspects of interactive systems (Damianos et. al 2000), however we present two possible ways of tracking what is being displayed on the screen. The first is to collect interaction data (as described in the previous section) and then recreate the session from the original state of the software. The second is to capture the display output at regular intervals using some sort of video format (digital or analog). These two approaches can be seen as analogous to the two approaches to coding that Nyerges et. al (1998) describe, as discussed earlier.

We have chosen to begin by implementing the first approach. If the initial state of the software is saved, the interaction log data can simply be used to fire off events, and replay the session. Furthermore, if the log data have been chunked as suggested above, the analyst should be able to replay the session in meaningful units, pausing where necessary, and perhaps fast-forwarding through tasks that are not of interest.

For example, if an analyst were to identify a certain type of high-level operation that seemed to be of particular interest. The analyst could use a query to discover that ten different subjects had executed that operation 53 different times in the process of testing a software tool, and could then go back to the data and quickly navigate to those key operations. He or she could then play back those sections of the log files and carefully examine what the user could have seen on the monitor, and compare that to what the user was talking about and doing.

This approach to analysis is essentially exploratory in nature. The goal is to find patterns in the data, and generate hypotheses that can be addressed in more rigorous testing. While this section has mainly dealt with the data that we propose to collect, and how we would handle this data, the remainder of this paper briefly describes an environment that would support this type of exploratory analysis of such data.

3.2 Exploratory visual analysis of data

Analysis of qualitative data has been characterized as an exploratory task (Wolcott 1999). Both the QDA literature and the cartographic literature note that exploratory data analysis is a highly iterative task (Coffey and Atkinson 1996 and MacEachren and Kraak 1997). Thus the system that we are designing needs to support two modes of use (at the highest conceptual level). The first mode involves exploratory tasks that provide summary representations of the three types of data that are discussed in the previous section. The second is continued analysis of that data, in the form of continued coding and recoding of the data, and refinement of the coding structure.

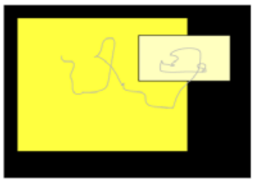
3.2.1 Exploratory Capabilities

There are a virtually unlimited number of exploratory capabilities that could be implemented in a system designed for analysis of the data that we have described above. Among these, there are a number of tasks that are common for tabular data, such as standard SQL-like queries, summaries such as counts and cross-tabulations, and summary statistics, which we plan to implement or leverage from existing database technologies for this system. In addition, there are a number of common navigation tasks associated with time-series data, such as animation (including standard VCR-type controls) and timeline plots, which will also be needed (some of which have already been implemented). However, the key contribution of this environment will be the interactive visual representations of the data. These will include standard exploratory data analysis tools, such as linking, focusing, and brushing (see MacEachren and Kraak 1997).

3.2.1.1 Interactive Visual Representations

The key to our proposed approach to analysis is the ability to make connections between interaction logs, verbal protocols, and what is being displayed on the screen. Therefore, we propose three highly interactive linked visual representations of these data sets.

Watcher window – The watcher window is designed to summarize what is being displayed on the screen. While we have already created tools that allow an analyst to animate a user interaction session, in a multi-representations analysis environment it is difficult to use the entire computer monitor to replay a session. One way of getting around this is to create a virtual window summarizing what is being displayed in the user interaction session. MacEachren et. al offer one example tailored to collaborative geovisualization environments, and call it a "watcher window" (MacEachren et. al 2001).



A watcher window with thematically shaded windows and a mouse trail.

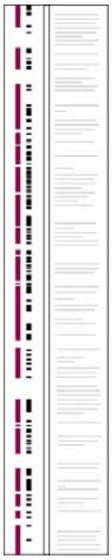
The watcher window would allow an analyst to see what windows are active, and where the mouse is at any given point in the interaction log. Furthermore, it can be animated or linked to a timeline such that moving to any location on the timeline will change the summary view of the watcher window. It could also be used to view a series of interactions in a static view. For instance, different window frames in the watcher window could be thematically shaded to indicate the percentage of time that they were active during a duration of the interaction log. In addition, mouse tracks could be used to show where the mouse was moved and clicked. The durations being analyzed could be non-continuous in time as well. For instance, if a user executed five brushing operations in their session, the watcher window could simultaneously display all of those operations (differentiated thematically using a visual variable such as color), and thus search for patterns in the use of brushing.

Finally, if an analyst were interested in interactions that took place in a specific region of the screen, he or she could "brush" that area, causing those interactions to be highlighted on the timeline.

Timelines with **Parallel Data Streams –** Standard timelines show events as points and lines along a continuous axis. In this system, we will implement timelines that simultaneously display interaction log data and user talk-aloud protocol data. The interaction data can be represented as points and durations on the timeline, and the verbal protocol data can simply be represented as lines indicating the density of verbal descriptions (see Robertson and Mackinlay 1993 for an example of summarizing text in this manner).

Timelines will allow interactive navigation to specific events. Timelines will also facilitate the selection of durations and non-contiguous sections of time. This can be useful for coding interactions. For instance, an analyst could select a series of low-level interactions by clicking and dragging the mouse over a duration on the timeline. Selecting these interactions could prompt the watcher window to display the mouse movements and the sequence of interaction with windows for that duration. Then the analyst could code the selected duration on the timeline with a context menu (right-click). This would provide an intuitive visual environment for analysis of interaction logs.

The data collected through talk-aloud protocols could be thematically mapped based on content, or coding (as per Miller et. al 1997), and expanded in a "lens" by either moving the cursor over, or clicking on a specific section of the text (see Rao and Card 1994).



The figure on the right is an example of a timeline with interaction log data on left, and verbal protocol data on the right. The interaction log data has both low-level interactions (in black) and higher-level operations (in maroon).

Tree diagrams of hierarchical coding – These diagrams will show relations between conceptual models and specific interactions or study participants' statements. The root codes of the tree diagram are the most abstract concepts, and are linked to more concrete concepts that are then linked to individuals in the data set. Highlighting a branch of the tree highlights all sub-branches, twigs, and leaves that come off of that branch (see MacDougal 1992). Furthermore, linking will allow sections of the timelines coded using those concepts to be highlighted, and the watcher window to form summary representations of all of those interactions.

3.2.2 Current State of Development and Future Directions

The GeoVISTA center has developed a computational workbench called GeoVISTA Studio (referred to here as "GVStudio") that supports the modular construction of interactive geographic visualization and exploratory data analysis (EDA) tools through a graphical programming interface (Gahegan et. al 2000). GVStudio is written in Java and supports any well-formed Java Bean as a module that can be immediately plugged into the visual programming environment. A number of geocomputational and representational modules have already been implemented, and research is currently being conducted using GVStudio for both the development of new visualization and EDA techniques, and the application of visualization and EDA techniques to specific research questions.

The visual representations described in the previous section will be developed as Java Beans, to be used in GVStudio (although they could also function as modules for other Java-based APIs). We have already implemented a module supporting the collection of interaction log data, and the ability to replay, and navigate through a recorded interaction log. This module can be added to any network of modules that is being used in GVStudio, and will immediately capture interactions at a low level of granularity. The module also provides a data structure through which a user can construct higher-level operations by aggregating these low-level interactions.

The next step in our development is to generate initial visual representations of the interaction log data, and provide direct links from those representations to editing tools, allowing users to dynamically define conceptual operations at a level that is sensible for specific types of analysis. We plan to follow this, then, by beginning to address tools for dealing with transcripts of talk-aloud protocols.

4 Conclusion

This paper has argued that there is a clear need for further development of methods and tools for the evaluation of interactive visualization software environments. Cartography is rapidly producing more and more such environments, and—with its long history of formal and informal evaluation of visual representations—should be able to provide a strong contribution to this effort. The ideas and software tools presented here make a start in that direction.

Work reported here was supported in part by the U.S. National Science Foundation (dgQG Project, grant #EIA9983451).

References

Bayer, S., L.E. Damiano, R. Kozierok, and J. Mokwa (1999). The MITRE Multi-Modal Logger: its use in evaluation of collaborative systems. ACM Computing Surveys, 31(2): Article No. 17

Buttenfield, Barbara P. (1999). Usability Evaluation of Digital Libraries. Science and Technology Libraries 17:3/4, 39-59

Chen, Chaomei and Mary P. Czerwinski (2000). Empirical evaluation of information visualizations: an introduction. International Journal of Human-Computer Studies. 53, 631-635

Coffey, Amanda & Paul Atkinson (1996). Making Sense of Qualitative Data: Complememntary Research Strategies. London: Sage

Damianos, Laurie E., Jill Drury, Lynette Hirschman, Jeff Kurtz, Beatrice Oshika & Tari Fanderclai (2000). Evaluating Multi-party Multi-modal Systems. A MITRE Technical Paper: http://www.mitre.org/support/papers/tech_papers99_00/damianos_evaluating/index.shtml

Denzin, Norman K. & Yvonna S. Lincoln (1998). Strategies of Qualitative Inquiry. London: Sage
Dey, Ian (1993). Qualitative Data Analysis: a user-friendly guide. London: Routledge
Gabbard, J.L., D. Hix, and J.E.I. Swan, 1999. User-centered design and evaluation of virtual environments. IEEE Computer Graphics and Applications, Nov/Dec: 51-59. for related ideas in the context of virtual environment design
Gahegan, Mark, Masahiro Takatsuka, Mike Wheeler and Frank Hardisty (2000) GeoVISTAStudio: a geocomputational workbench. GeoComputation 2000. Kent, UK
Glaser, Barney G. & Anselm L. Strauss (1967). The discovery of grounded theory: strategies for qualitative research. Chicago: Aldine de Gruyter
James, Jeffrey M. & Penelope M. Sanderson (1991). Heuristic and statistical support for protocol analysis with SHAPA Version 2.01. Behavior Research Methods, Instruments, & Computers 23:4, 449-460
Kaugers, Karlis (1998). Integrated Multi Scale text Retrieval Visualization. Proceedings of Chi '98, April 18-23; pp. 307-308
Klein, Heinz K. & Michael D. Myers (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. MIS Quarterly 23:1, 67-93
Miller, Nancy E., Beth Hetzler, Grant Nakamura & Paul Whitney (1997). The Need for Metrics in Visual Information Analysis. In Workshop on New Paradigms in Information Visualization and Manipulation in conjunction with the Sixth AXM International Conference on Information and Knowledge Management, November 13-14, Las Vegas, Nevada: ACM Press
MacDougall, E. Bruce (1992). Exploratory Analysis, Dynamic Statistical Visualization, and Geographic Information Systems. Cartography and Geographic Information Systems 19:4, 237-246
MacEachren, Alan M. & Menno-Jan Kraak (1997). Exploratory Cartographic Visualization: Advancing the Agenda. Computers and Geoxciences 23:4, 335-343
MacEachren, A.M., I. Brewer, and E. Steiner, (in press). Geovisualization to Mediate Collaborative Work: Tools to Support Different-place Knowledge Construction and Decision-Making, 20th International Cartographic Conference. ICA, Beijing, China, August 6-10, 2001.
Mark, David M., Christian Freksa, Stephen C. Hirtle, Robert Lloyd & Barbara Tversky (1999). Cognitive models of geographical space. Int. J. Geographical Information Science 13:8, 747-774
Morse, E., M. Lewis & K.A. Olsen (2000). Evaluating visualizations: using a taxonomic guide. International Journal of Human-Computer Studies 53, 637-66
Nielsen, J., (1993). Usability Engineering. Academic Press, New York.
Nyerges, Timothy L., T.J. Moore, Robb Montejano & Marcie Compton (1998). Developing and Using Interaction Coding Systems for Studying Groupware Use. Human-Computer Interaction 12, 127-165
Rao, R. , and S. K. Card. (1994). The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In Proceedings of ACM CHI'94, pages 318--482.
Robertson, George G. & Jock D. Mackinlay (1993). The Document Lens. In Proceedings of the ACM Symposium on User Interface Software and Technology, November 3-5, ed. ; pp. 101-108
Schneiderman, B., 1998. Designing the User Interface - Strategies for Effective Human-Computer Interaction. Addison-Wesley Longman, Inc., Reading.
Slocum, T.A., C. Blok, B. Jiang, A. Koussoulakou, D.R. Montello, S. Fuhrmann and N.R. Hedley (2001). Cognitive and Usability Issues in Geovisualization. Cartography and Geographic Information Science, 28(1): 61-75.
Strauss, Anselm L. & Juliet Corbin (1990). Basics of qualitative research: grounded theory procedures and techniques. Newbury Park, CA: Sage
Wolcott, Harry F. (1999). Ethnography: A way of seeing. London: Alta Mira Press