VISUAL ANALYTICS FOR LARGE SCALE ETHNOGRAPHY: EXPLORING COMPUTATIONAL AND ETHNOGRAPHIC ANALYSIS TECHNIQUES TO BETTER UNDERSTAND THE OCCUPATION AND UTILIZATION OF THE BUILT ENVIRONMENT

by

Steven McMillan Danilowicz

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degrees of Master of Science in
Architecture & Information Technology

Charlotte

2015

Approved by:

____
Dr. Richard Souvenir

____
Professor Eric Sauda

____
Dr. Donna Lanclos

ABSTRACT

STEVEN MCMILLAN DANILOWICZ. Visual analytics for large scale ethnography: exploring computational and ethnographic analysis techniques to better understand the occupation and utilization of the built environment.
(Under the direction of DR. RICHARD SOUVENIR)

Observational data within Post Occupancy Evaluations (POE) has had the problem of being deemed as 'unreliable' due to the fact that it is not based on quantifiable, empirical evidence. Even though ethnographic observation is quite effective and has a method, the common misconception is that it is just someone watching and drawing conclusions. What I propose is a system that can turn some of the ethnographer's observational data into empirical data that can be better incorporated into traditional POE reports. The data becomes more credible in the eyes of those unfamiliar with ethnographic observational practices, and can be used to benefit current and future building designs.

Such a system utilizes principles of Computer Vision and Machine Learning to develop a means for collecting and synthesizing observational data. The system will use a camera array installed in a space to track the movement of the people within. Additionally, we will enlist the help of ethnographers to teach the program to look for 'meaningful interactions' of the people within the space. Based on these notable events, the program will be able to draw data from like events, store the data, and use it when examining future data. This system will allow for data collection to occur continuously within a building where such a system is installed, providing a continuous stream of information to be used in Evidence Based Design. In this way, the evaluation team will be able to present a new dimension of data in their POE for the architects and owners to consider.

The following thesis addresses background information on the structure of POEs, Ethnographic methods of observation, and proposes a preliminary User Interface for inputting information as well as a method for identifying 'meaningful interactions'.

INTRODUCTION

Architects have designed buildings with the intent of orchestrating the way that people move about and interact within their buildings. The architect uses techniques such as circulation control and program adjacencies to govern how people move about in the building. Unfortunately, one cannot know for sure if these strategies and designs are successful until the building has been constructed and occupied. After its completion, an architect may conduct what is known as a Post Occupancy Evaluation (POE) of the building. POEs are normally done approximately two years after the buildings completion.

In a POE, the evaluators may be social scientists that practice a method of data collection that has foundations in ethnography, or are themselves ethnographers. The type of evaluator may depend on the type of evaluation. For example, an evaluation that is only collecting survey data would only require someone that could formulate surveys. This evaluator would not necessarily need the observational skills of a trained ethnographer. Paul Dourish, in his opening chapter to Ways of Knowing, says that ethnography is "the deliberate attempt to generate more data than the investigator is aware of at the time of collection." Ethnography is a process of data production, an active process that the researcher is involved in while in the field. Ethnography is not a spectator sport, as it may seem to the casual observer, but is in fact a highly involved activity that requires constant engagement within the setting. This concept brings up the question of whether or not the ethnographer interferes with the scene that he or she is observing simply by being there. Dourish answers, yes, but all of the participants in the setting also change and affect the scene.

Within POEs, evaluators with no ethnographic training tend to utilize tools such as surveys and focus groups to gain data on the success of the space and the way that people perceive it. These initial data sets are sometimes later augmented by observations conducted by ethnographers, but due to time constraints, these observations can only take place over a certain amount of time before the ethnographer must step back and synthesize the gathered data. As a result of this restriction, the observed data are sometimes extrapolated over the course of a year to form a projection of the conditions for the building.

What I propose is a system that aids evaluators and ethnographers in their observation data collection portion of the POE. An observation system based on Computer Vision and Machine Learning principles could allow an ethnographer to 'teach' the system to look for meaningful patterns of interaction between occupants and their environment in order to form a long-term data collection method. The following thesis addresses background information on the structure of POEs, Ethnographic methods of observation, and proposes a preliminary User Interface for inputting information as well as a method for identifying 'meaningful interactions'.

TABLE OF CONTENTS

LIST OF FIGURES vii

CHAPTER 1: THESIS DISCOURSE 1

- 1.1. Post Occupancy Evaluations 1
- 1.2. Ethnography in Post Occupation Analyses 4
- 1.3. System Design Strategies 7
- 1.4. Computer Science: Precedents and Methods 9
- 1.5. Potential Applications 14
- 1.6. Conclusions 16

CHAPTER 2: CURRENT STATE OF VALSE 17

CHAPTER 3: INTERFACE PRECEDENTS 21

CHAPTER 4: VALSE INTERFACE DESIGN 30

- 4.1. Visualization Panel 32
- 4.2. Event Timeline 35
- 4.3. Event Creation 40
- 4.4. Element Management 45
- 4.5. New and Multiple Windows 48

CHAPTER 5: CONCLUSIONS AND FUTURE WORK 51

BIBLIOGRAPHY 52

LIST OF FIGURES

FIGURE 1: Post Occupancy Evaluations 2

FIGURE 2: MERL forensic surveillance system. 10

FIGURE 3: Small group tracking. 11

FIGURE 4: VALSE camera array. 17

FIGURE 5: VALSE camera views. 18

FIGURE 6: VALSE prototype. 19

FIGURE 7: Meaningful Event identification. 20

FIGURE 8: ELAN interface. 21

FIGURE 9: VCode interface. 22

FIGURE 10: Morae interface. 23

FIGURE 11: Rhinoceros interface. 24

FIGURE 12: Adobe Illustrator interface. 25

FIGURE 13: Dot trail pathway representation. 26

FIGURE 14: Vector trail pathway representation. 26

FIGURE 15: Ekahau interface. 27

FIGURE 16: CFD Simulation heat map. 28

FIGURE 17: iCal event creation interface. 29

FIGURE 18: VALSE wireframe interface. 31

FIGURE 19: Vector trails. 32

FIGURE 20: Dot symbol trails. 32

FIGURE 21: Combination dot and vector trails. 32

FIGURE 22: Population heat map. 33

FIGURE 23: Snapshot view. 33

FIGURE 24: Event timeline. 35

FIGURE 25: Datetime menu. 36

FIGURE 26: Event timeline examples. 37

FIGURE 27: Unfocused timeline. 39

FIGURE 28: Focused overall timeline. 39

FIGURE 29: Focused timeline panel. 39

FIGURE 30: Spatial proxemics radii. 41

FIGURE 31: Simple event - Passing through. 42

FIGURE 32: Simple event - Sitting. 42

FIGURE 33: Compound event - Checking in and waiting. 43

FIGURE 34: Simple event - Conversation. 43

FIGURE 35: Simple event - Meeting. 44

FIGURE 36: People layer sidebar options. 46

FIGURE 37: Event layer sidebar options. 47

FIGURE 38: Two interaction windows. 48

CHAPTER 1: THESIS DISCOURSE

1.1. Post Occupancy Evaluations

As it stands, Post Occupancy Evaluations alone are no longer enough in this age of technology. With technology becoming more and more ubiquitous, architects and designers need to start thinking about the ways in which it can be incorporated in to the design of the building. In this case, means in which to give an evaluator constant feedback about the ways in which the building is being used is an example of how architects can start to think about this incorporation. Currently, evaluators are focusing on hard data that is easily collected through surveys and focus groups that are the mainstay of Indicative and Investigative POEs. What is being missed here is the opportunity to incorporate elements of the more in depth Diagnostic POE into these practices. Although Diagnostic POEs take much longer, they yield far more data than the quick glance Indicative and Investigative methods.

Most of the time, conducting a POE yields results in these areas (Preiser, 20):

- Health and safety problems
- Security problems
- Leakage
- Poor signage and wayfinding issues
- Poor air circulation and temperature control
- ADA issues
- Lack of storage
- Lack of privacy
- Hallway blockage
- Aesthetic problems
- Entry door problems with wind and dirt accumulation
- Inadequacy of designing space for equipment
- Maintainability of glass surfaces

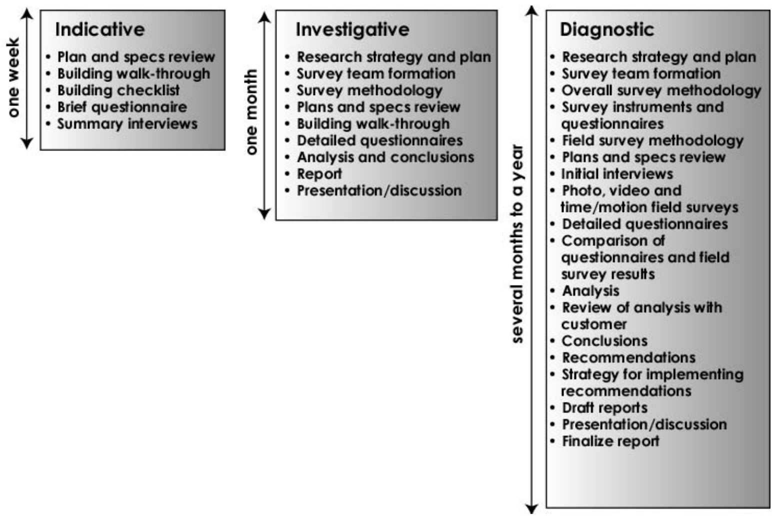


FIGURE 1: Post Occupancy Evaluations. This diagram illustrates the relation of complexity between the three different types of Post Occupancy Evaluations

The majority of these issues can be addressed without any sustained observational interaction on the part of the ethnographers. Surveys and interviews with those that work and interact in the building on a consistent basis will be able to give the proper type of feedback the ethnographers can use to create the assessment of the building in this area. However, things that are missing from this list are elements such as:

- How efficient is the placement of equipment in the space?
- How does the layout of individual rooms affect occupant's productivity?
- How does the layout of rooms on a building floor affect an occupant's productivity?
- How does the layout impact travel time for occupants (this is addressed tangentially and only in the negative through wayfinding above)?
- How well do the workers within the space operate when interacting with visitors or customers?

Observational questions such as these demand more attention from an ethnographer and may be able to lead to deeper answers that can positively affect the ways in which architects design buildings of this type in the future.

Issues that lie with this type of observation and evaluation are the fact that this type of study is perceived as objective in nature and is not necessarily hard data that can be directly used in Evidence Based Design (EBD). EBD uses the data gathered from building statistics (cooling loads, electrical consumption, light levels, air particle counts, etc.) to create a means in which architects and engineers can access a pool of data when designing future buildings of the same type. By incorporating a more “hard data” category of observation into this EBD strategy and allow for an easier, less time consuming method of recording, designers could use valuable observation data more in their future design decisions.

1.2. Ethnography in Post Occupation Analyses

When examining the ways in which ethnographers perform the duties of a POE, we need to ask the following questions:

1. How does ethnographic field research actually work, and
2. How are misconceptions associated with observational data collection leading to the loss of this data collection within POEs?

To begin, ethnographic analysis is based around two elements working in concert:

1. Observation: Seeing how people act differs from what they may say they do.
2. The application of Social Theories and Anthropological Methodologies.

Ethnographers are their own instrument, and must be finely tuned with training and experience. In this sense, ethnography is a fairly refined field of study, and relies heavily on the sensibility of the trained observer to add constraints to their observations in order to apply a systematic approach to their findings. It is this methodology of observation and way of thinking that needs to be considered and applied to a computer vision and machine learning based system in order to achieve a level of consistency that allows for the data collected to be usable.

But this also raises the question of who is performing the evaluations. In the case of an Indicative Evaluation and an Investigative Evaluation, the evaluators may be people that are well versed at crafting surveys and conducting interviews. Perhaps these people are skilled social scientists or counselors of some nature. In some cases, there may also be the misconstrued idea of "it's just observation, I can do that myself." This mentality leads to poor observations and skewed data, and falls within a series of misconceptions about ethnography that affect designers trying to utilize that sort of skill set within their research. They are as follows:

1. Anyone can do ethnography – it's just a matter of common sense.
2. Being insiders qualifies people to do ethnography in their own work setting.
3. Since ethnography does not involve preformulated study designs, it involves no systematic method at all – "anything goes."

4. Doing fieldwork is just chatting with people and reporting what they say.
5. To find out what people do, just ask them!
6. Behavioral and organizational patterns exist "out there" in the world; observational research is just a matter of looking and listening to detect these patterns.

These are fairly common misconceptions due to the fact that to the casual observer, someone performing an ethnographical study is simply watching people. These are methods and techniques that are honed over time and turn the ethnographer into a finely tuned instrument. This is the same as in any discipline. The more a person practices and learns the means and methods of the profession, the easier it seems to be to the outside observer.

Here, we can offer some corrections to these misconceptions:

1. Anyone can do ethnography – it's just common sense: Ethnographers use more than common sense when observing. They must identify and problematize things that the insiders take for granted in order to fully analyze their situation.
2. Being and insider qualifies people to do ethnography in their own work setting: Building on #1, an ethnographer is most effective when he or she is an outsider with considerable inside experience. The ethnographer may use this experience to enhance their observations and help determine key points to focus on when observing.
3. Since ethnography does not involve preformulated design studies, it involves no systematic method at all - "anything goes": Just because ethnography isn't conducted in a controlled environment doesn't make it an "anything goes" study environment. There are still means and methods that are used to study groups. Adjusting these methods mid study to react to the inside group takes considerable skill.
4. Doing fieldwork is just chatting with people and reporting what they say. The social scientists job is to analyze what people say, not merely record it. There are meanings and layers within what people choose to say and the context that they say it in.
5. To find out what people do, just ask them! Talking with people with only get you half of the information that you would need as an ethnographic observer. People tend to say one thing and do another when describing what it is they are doing. Or they may leave out key steps. It is the ethnographer's job to pick up on these through observing.
6. Behavioral and organizational patterns exist "out there" in the world; observational research is just a matter of looking and listening to detect these pat terns. Simply watching for patterns is not as efficient as field observation where context can be taken into account. This is analogous to a doctor performing a diagnosis based off of looking at a picture of a patient.

As designers of a system that is going to examine the ways in which people interact with the space and each other in the space, all of the before mentioned points need to be addressed when designing the system. The most poignant part of this critique is that the proposed system is essentially trying to do just what the last misconception above is pointing out: find behavioral patterns and extrapolate data from them. It is because of this point that we cannot remove the ethnographers from observational POEs. Observation relies on context, and computer based pattern detection is not sophisticated enough for the inclusion of context yet. Therefore, our system will be limited in its application. As a result, this type of system would take the smaller observational tasks off of the hands of the ethnographer and allow him or her to focus more on in depth observations such as interpersonal relations.

If the system can be built in a way that gives video data more dimensionality and allows for a more intimate viewing of the information, ethnographers may have a more in depth look at past data than they would have with simple video collection. Ethnographers may be able to pick up on patterns that they could have missed otherwise.

1.3. System Design Strategies

As we consider the elements of both ethnographic practices along with the state of POEs, we can begin to construct the framework for a system that will aid ethnographers in their observations. The ultimate goal of such a system is to use Machine Learning and Computer Vision to track and draw 'meaningful interactions' from an environment based off of an ethnographer's expertise. When developing a new system, we must consider what Akdemir et al state:

> "Structural approaches usually rely on hand-crafted models from experts or analysts, hence they are intuitive and can be related to the semantic structure of the activity. Since, these approaches rely on a domain expert to provide the activity semantics, it is useful to create a standardized knowledgebase from which to draw upon."

We need to establish a clear division of labor based on disciplinary expertise in order to create a robust program. In the case of this observational, ethnographic system, ethnographers must be used to create a vocabulary of actions to use in the machine learning aspect of the system design. Using their expertise, an ethnographer must first visit a site, clearly establish what types of behaviors constitute 'meaningful interactions,' and work with an evaluator to translate these interactions into a language the computer will understand. Akdemir goes on to state five points to structure the criteria for creating such a vocabulary: clarity, coherence, extendibility, minimal encoding bias, and minimal commitment.

1. Clarity: A vocabulary should convey the meaning of all conceptualizations unambiguously.
2. Coherence: A vocabulary should be coherent and allow for meaningful inferences to be drawn that are consistent with definitions and axioms.
3. Extendibility: The design of the vocabulary should take into account future extensions without a need for revising definitions.
4. Minimal Encoding Bias: A vocabulary with minimal encoding bias represents the real world with minimal skewing of the data collected to fit into the system framework.
5. Minimal Commitment: A vocabulary makes as few assumptions as possible about the domain being modeled.

These rules are the basis for creating the observational vocabulary with which ethnographers would teach the system what to look for through a User Interface (UI). We are designing algorithms to look for these specific patterns in video data collected through a camera array. For privacy reasons, the video data is broken down into vectors of the people's movement through space with no personal identifiers. Ethnographers and evaluators should be able to access such gathered information through the UI and input new elements into the vocabulary of the system as they become pertinent to the types of data the observers wish to collect.

1.4. Computer Science: Precedents and Methods

Implementing a system such as this requires the development of the system in the areas of Computer Vision, Context Aware Computing, and Machine Learning. Each of these fields contributes to the system in various ways. Firstly, Computer Vision is the field of computer science that studies the ways in which computerized systems use sensors, specifically cameras and motion sensors, to see and interpret their surrounding context. This is one way in which it can feed into Context Aware Computing, another principle that uses the environmental context of the computers environment to inform decisions. Finally, Machine Learning can build off of these principles in order to form patterns and identify the way that users interact in the space and pull from as well as contribute to an established ontology of actions.

Computer Vision and Machine Learning concepts have been built into systems prior to this one in ways that use technology in a manner similar to the goals that I have set forth in this paper.

Ivanaov et al's paper, "Visualizing the History of Living Spaces," covers the development of a system that was designed to help analyze video data based on the traffic of people within range of a camera. This was done with a series of sensors imbedded in the ceiling along with the security cameras in an office building. In this experiment, when a sensor was tripped, the camera would begin to record until the sensor no longer detected movement in the area. This cuts down on video footage of the cameras line of sight, and creates an index within a database that is represented both temporally and spatially. Tracks of the way that people moved through the space are recorded as dashes on a "piano roll" interface indicating the time that they were active in a space.

The systems interface allows a user to interact and analyze the data based on a variety of visual displays. This may be a map with the activity highlighted by zones, or graphs displayed in the "piano roll" sense that show a week's worth of data at a time. By using these systems in conjunction with each other, users can sift and sort through the data to track a single person or moving object through the space. However, this technique begins to break down in high volume areas because of the ambiguities of the people and objects moving in the same space simultaneously.

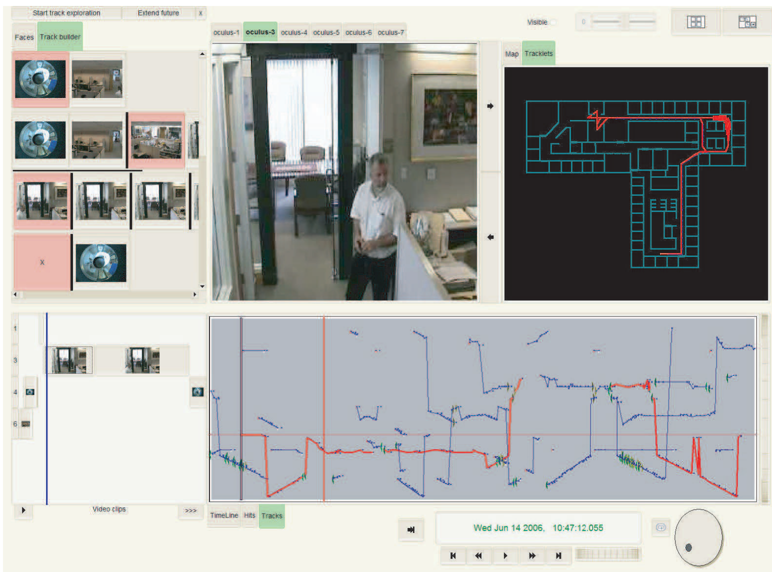


FIGURE 2: MERL forensic surveillance system. This is the graphic user interface proposed by Ivanov et al.

This type of system is a wonderful example of the type of interaction that an end user should have with the proposed system. An interface that allows the user to move back in time and analyze the data for events and occurrences would be perfect, because it allows for an ethnographer to begin to form patterns and program the machine to record similar events in the future. The system also abstracts data on a floor plan, and only records when there is movement in the cameras sensing range. Both of these are important to consider when examining the issue of privacy in a system that collects video data.

Ge et al's paper, "Automatically detecting the Small Group Structure of a Crowd," covers the development of a system that applies a tracking algorithm to crowd groups in order to determine the structure of the crowd. The assumption of this paper is that crowds are not made up of individuals, but rather small groups of people. Thus, only in extreme cases will you see crowds that are made up entirely of individuals.

The model that is applied to the crowd detection deals with the velocity that the group members are each traveling at along with the travel vector and compares them with each other. Additionally, it measures the proximity of the members in the group to each other. This helps when determining a group of people that are not moving, but may be having a conversation while standing still.

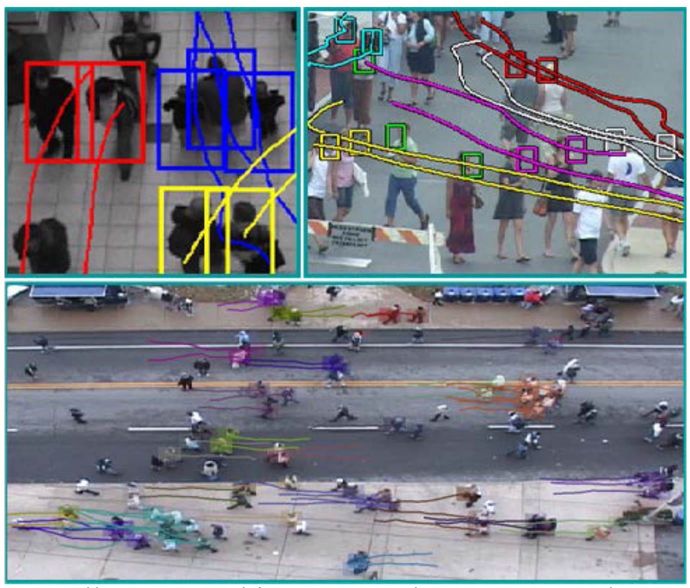


FIGURE 3. Small group tracking. Ge et al's system can determine the various small groups of people that make up pedestrian traffic

Combined with the previous example, person tracking and group identification are shown to be viable features and applicable to the proposed system. Using a technique such as the one Ge et al mention in combination with a system that Ivanov et al mention could develop into a very strong system of analysis and interpretation of the ways that people interact with both each other and their environment. Both of these are invaluable pieces of data to an ethnographer, and if they can be recorded through an interface such as the previous two, then observational analysis may be able to make its way into Indicative and Investigative POEs.

Context Aware Computing also plays a role in this development. As designers, we need to follow a set of guidelines in order to ensure that a Context Aware system makes as few mistakes as possible. There are three foundational aspects to consider when designing a context aware system such as this: Accuracy of information, confidence in action (in the case of this system, assigning a tag to a person), and the potential consequences of the action (especially whether or not it's correct). Data gathered from any sort of system will be inherently flawed to some degree. The important thing to keep in mind here when designing this type of system is to allow for a certain amount of error tolerance. Systems must be able to account for the accuracy of the inference before deciding which action to take. In the case of a visually based system, it will need to incorporate some type of error estimator before analyzing and labeling a person within a space, or action that a person is taking.

In the case of actions, it is necessary to consider the consequences of the system's actions. Are the actions intrusive or not? Do they need to be intrusive based on the context of the user? There are four categories of action, beginning at the least intrusive, and moving to the most:

1. Tagging actions focus on attaching meta-information to some data based on context information for later use. The user can then at a later point use this information to e.g. categorize captured images.
2. Displaying actions focus on organizing and displaying context awareness information to the user to enable decisions based on context aware information (context-mediated awareness)
3. Helping actions focus on preparing some actions that the computer thinks the user is going to need, but it is still the user who initializes the actions.
4. Executing actions focus on executing specific actions automatically without involving or asking the user in the process.

The type of action (Tagging, Displaying, Helping, or Executing) depends on the system design and how confident the system is in the users action. By programming the computer to react to different actions in different manners, a context aware system could build upon a users experience and help the user by supplying him or her information or performing tasks automatically. The consequences of this programmed action need to also be taken into consideration before executing the prescribed action.

Unfortunately, context aware computing programs have trouble identifying the exact context of what a person may be doing in a space. Because the context is subjective, it is nearly impossible to determine the correct context for a user within a space without other clues. For example, a patient entering an operating room usually means that he or she is going in for a surgery. A doctor, however, may have multiple reasons for entering that same room. This idea of meaning is something that is lost on the computer and cannot be inferred from sensor data alone.

1.5. Potential Applications

Such a system has many potential benefits in various sites, but one of the better applications is inside of a healthcare environment. Here, the system has the capability to observe the staff, patients, and visitors as an ‘invisible observer’ and collect data that an ethnographer might otherwise skew from being present. Even though Dourish points out that an ethnographer’s presence is a part of the activity, this ‘invisible observer’ concept may reveal a different set of results seen in the data over an extended amount of time. An array of this nature would be able to aggregate a series of patterns that may take longer to develop than a single observation period conducted by an ethnographer.

In a hospital environment, such a system could be placed in a variety of environments such as:

- Waiting rooms
- Patient rooms
- Nursing stations
- Hospital corridors
- Operating rooms

Each of these environments yield a distinct variety of observable opportunities. Some of which may not be available to direct observation due to health and safety measures enforced by the facility.

Waiting rooms yield the obvious solutions of tracking where people sit, how they may congregate in a room based on social dynamics of family groups, clustering due to respect for privacy, amount of light, and air temperature. The system may also be able to determine how often people interact with the administrative staff or nursing staff and how people in the waiting room react to new comers in the waiting room.

Patient rooms can reveal how effective the room design is at allowing for patient mobility within the room, how the family or other visitors occupy it when seeing the patient, how easily equipment is moved in and out of the room and where it is placed, and how often patients may try to leave their beds on their own.

Nursing stations and hospital corridors placement may reveal how often nurses visit the station for medication and the amount of traveling that they do on a daily basis to and from the station.

Finally, operating rooms may be an area in which the ethnographer could have no access to when they are functioning. The system may be implemented to track how the surgeons and nurses move about the space and equipment. How well is the equipment placed in the room? For now, this is all speculation of potential opportunities for such a system in a healthcare environment. Future interviews with POE evaluators and ethnographers may reveal more opportunities for observation. As the system is used and opportunities arise, the flexibility should allow for more elements to be added to the vocabulary of the system.

1.6. Conclusions

Producing a system of this caliber will allow for an easy analysis of a large quantity of data that was previously more difficult to obtain. Using this system in tandem with traditional ethnographic observations and analyses will increase the depth of Indicative, Investigative, and Diagnostic Post Occupancy Evaluations through the addition of minor, but not trivial, data points about the people within the building. Our future endeavors into developing this system will begin to reveal an appropriate user interface, preliminary vocabulary of 'meaningful interactions' after interviewing ethnographers and POE evaluators and attending a POE, and finally, a prototype of the system itself. This study will also help to develop a methodology for understanding how social theories and anthropological methods used within ethnography work within the framework presented by Akdemir when designing a new system.

CHAPTER 2: CURRENT STATE OF VALSE

The VALSE system has been under development by Dr. Richard Souvenir and his team in the Video and Image Analysis Lab for the past few years. As it currently stands, the system has the ability to track people as they move through the space in which it is installed. The system is made up of a camera array installed within the ceiling of a space that feeds video data back to a computer. The VALSE program uses Computer Vision algorithms to calculate the positions of people within the space based off of triangulating their positions. Because there are many cameras in the array, the system can continuously track people as they move through the space, even if the person is obstructed from a cameras view. Here is an example of the camera layout of the VALSE test system installed in the UNCC Center City Building:

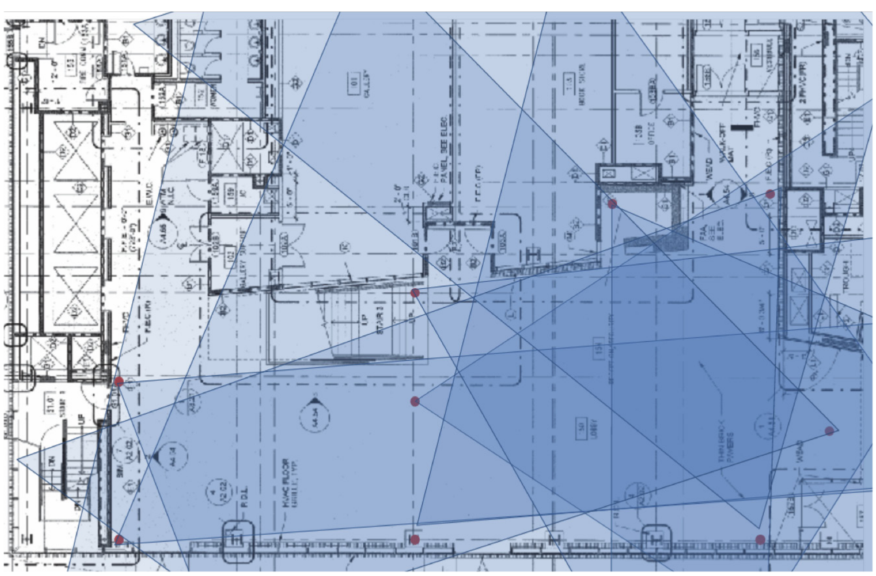


FIGURE 4: VALSE camera array. The layout of the cameras allows for tracking users in a space where their pathway may be obscured from at least one camera.

FIGURE 5: VALSE camera views. Views of the space as seen by the installed camera array.

As VALSE tracks people through the space, this infomation is compiled and shown as a series of tracks on a floor plan view of the space. This information is shown in conjunction with a timeline controller, population histogram, and heat map of the observation area. The timeline controller is the main method of interaction with the system, and has a pair of handles that can be used to focus the timeline and display a more narrow band of pathway information. This action affects the displayed pathways on the floorplan as well as the information in both the histogram and heat map.

Currently, the VALSE system is only capable of identifying where people are within a space and tracking their movements through it. Using a meeting as an example, we can identify the capabilities of VALSE as well as the prospective goals for the system. Step 1: Detection. The VALSE system has the basic ability to identify people within a space. In the case of the meeting, VALSE can identify the group of people sitting around the table at which they are meeting. Step 2: Tracking. The VALSE system has the ability to take the identified people within the space and then track their movements within the observed area. The system accomplishes this feat by utilizing the miltiple cameras

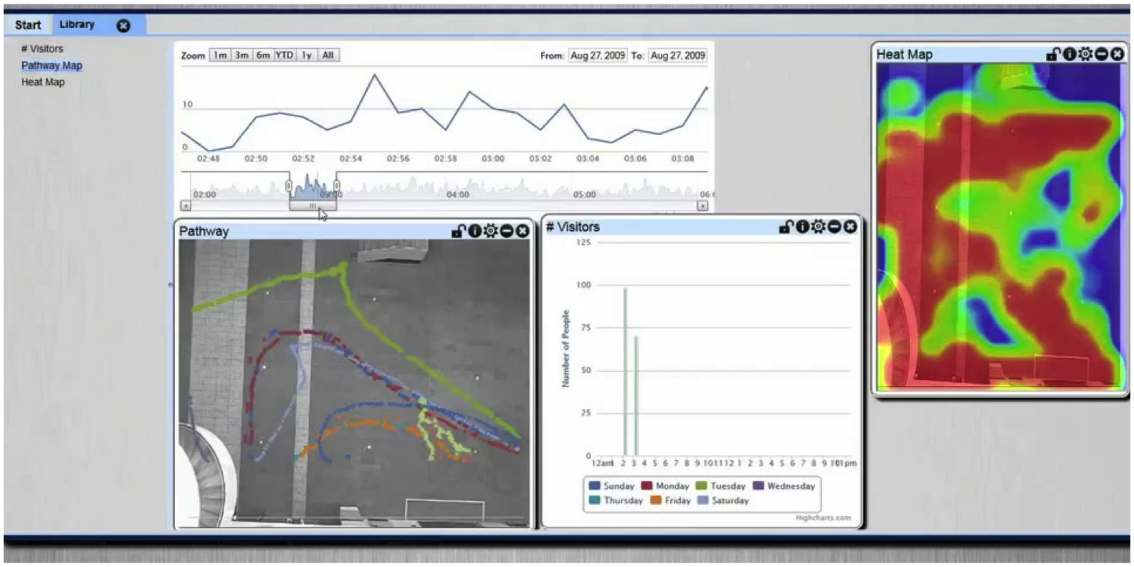


FIGURE 6: VALSE prototype. This prototype illustrates the initial system inter face as well as the basic components needed to interact with the gathered information.

observing the space and triangulating the position and trajectory of each person within. In the case of the meeting, VALSE is able to track the people as they move to the the table, and if they move around it. These first two steps signify the limitations of the system as it stands now. The following two steps are further goals for the system. Step 3: Low Level Action Recognition. Identifying the actions that a person or two people are taking is a complex Computer Vision task. VALSE cannot yet identify two people interacting on an close level such as the handshake that we see here. However, if we utilize permanent markers within the environment, and assign importance based on a persons proximity to that marker, we may begin to develop this activity recognition on a limited level. Step 4: Activity Recognition. This is a complex task that involves recognizing body language as well as proximity and location within space. As humans, we can tell that this group of people are having a meeting based on social cues and standards. Computers looking at this scene would not be able to deduce that. It is our ultimate goal that through Machine Learning and user interaction with the system, that we will be able to build a significant enough sample set of data for the system to deduce such an event.

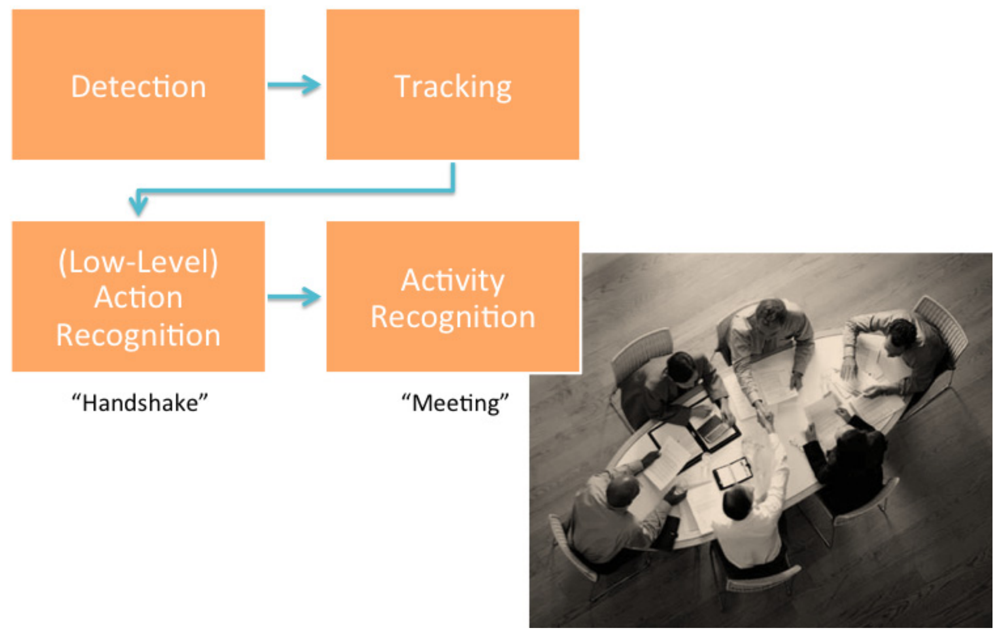


FIGURE 7: Meaningful event identification. The example of the meeting out lines the abilities and ultimate goals of the VALSE system in terms of event identification.

CHAPTER 3: INTERFACE PRECEDENTS

The VALSE system was created with the influence of other systems addressing similar issues. VALSE is an annotation software that deals with temporal and spatial identification for marking an event. To explore these techniques, I examined a variety of annotation software: ELAN, VCode, and Morae.

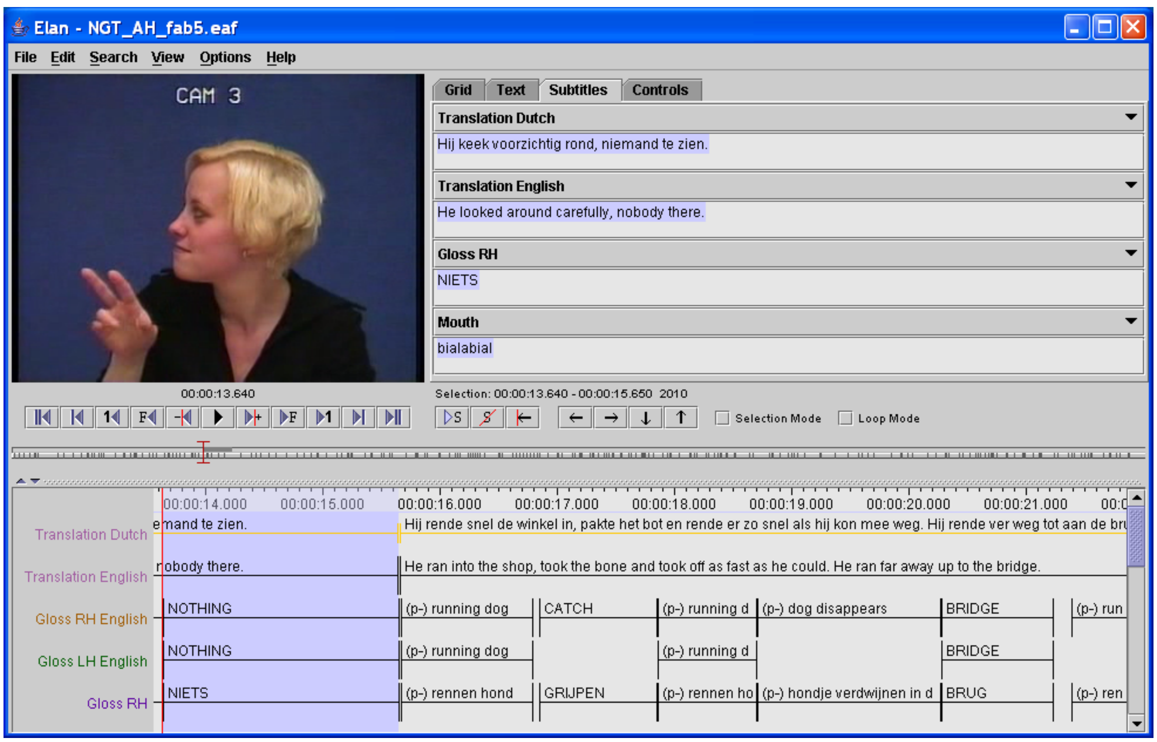


FIGURE 8: ELAN interface.

The ELAN annotation tool allows for a user to watch a video recording and make notes on what happens in the recording. A user may transcribe what is being said or make notes on actions taken by the people or person within the recording. This is accomplished by placing markers on the timeline panel that occupies the lower half of the window. These markers denote the start and stop point of the action, but do not have a marker on the recorded video denoting where it happens. Because the system is meant for an intimate scale of recording actions and speech, a simple note on the annotation will suffice. Aditionally, there is a set of timeline controls that allows a user to select an area of interst and loop on that area over and over again. This is helpful in this type of annotation especially when deciphering what a person is saying. This level of engagement will also be helpful in VALSE when identifying meaningful interactions performed in the observed area.

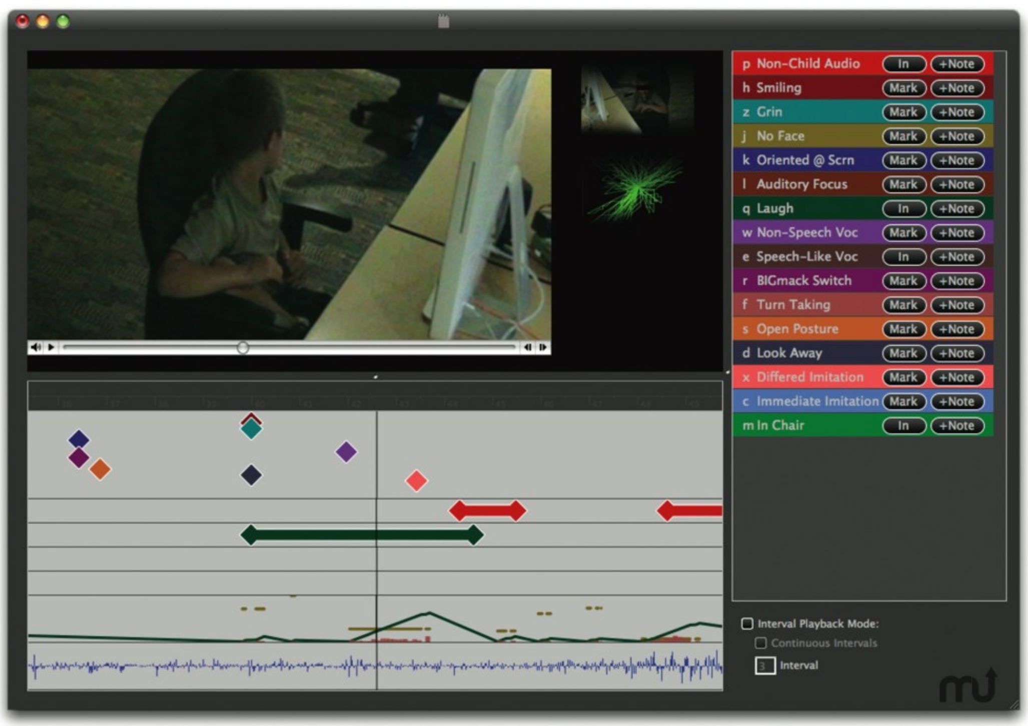


FIGURE 9: VCode interface.

The VCode interface is similar to ELAN in the way that it is also an annotation software used to add notes to a set of video data. This system is different in that it places the value of the video over the notation system. Here, we see that the size of the video is much larger than the one presented in ELAN, and that the notations on the timeline are symbols rather than words. In VCode, the events have been pre-defined on the right hand column and are represented as a marker, or extended marker depending on the type of event. This type of representation is more applicable to the VALSE system due to the fact that displaying the spatial data is important to the user when he or she is identifying meaningful events within the space.

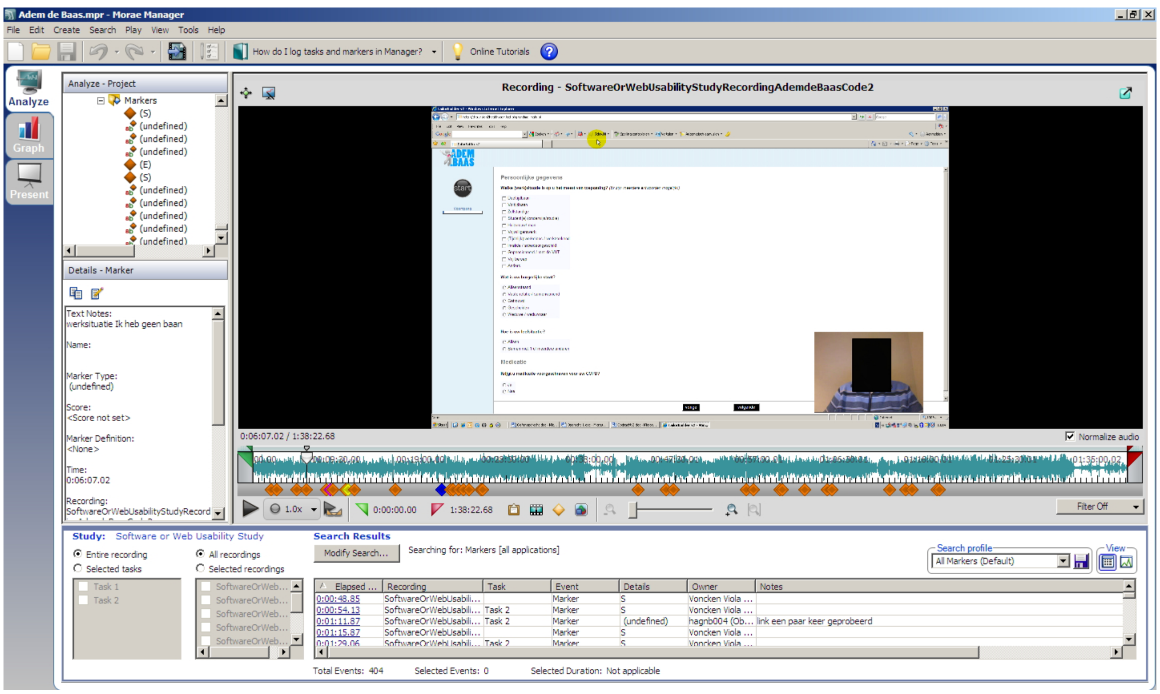


FIGURE 10: Morae interface.

Morae is an annotation software that is used when testing desktop interfaces on new users. The system records their mouse movements as well as what is displayed on the screen and records their facial expressions. Like ELAN and VCode, this system uses a timeline and markers to identify where events occur within the recording. The method of identifying and marking events is more similar to that described in VCode, but also takes advantage of the depth that ELAN uses. The events are denoted as diamonds on the timeline and are sorted on the sidebar by category. These events may also have notes attached to them and encompass a certain duration on the timeline. Morae also focuses on the display of the users screen far more than ELAN and VCode, something that can be incorporated in VALSE. In addition to the smilarities in the recording display and the notation on the timeline, Morae also uses the same timeline mechanics as our VALSE prototype. Two handles are located on either end of the timeline, and can be dragged to identify a more focused area of the timeline for deeper investigation.

In addition to annotation software, I also investigated the drawing tools of Rhinoceros and Adobe Illustrator. Since our evaluators may be people from an architectural office, incorporating commands and means of interaction from drawing software would be an easy, understandable transition from one platform to another.

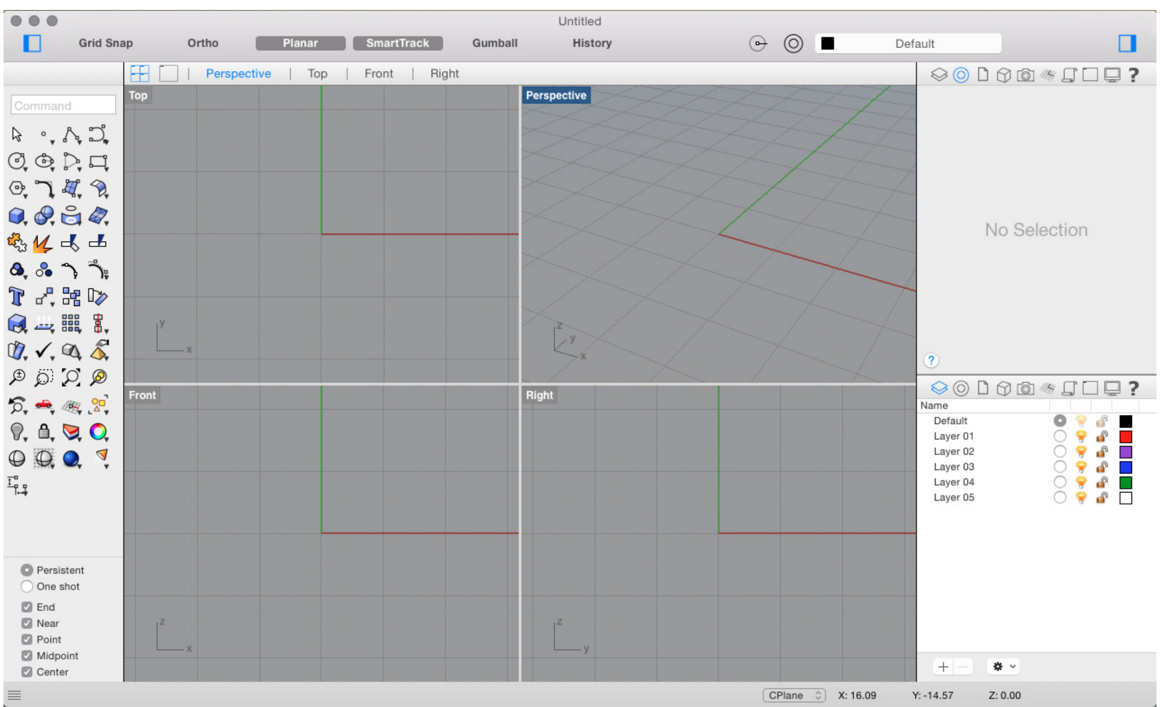


FIGURE 11: Rhinoceros interface.

Rhino offers the user the ability to draw shapes within an environment depicted in four quadrants. The user may utilize drawing tools on the left panel, and can edit and sort these elements within the layers panel seen in the bottom right of Figure 11. Interaction with the Visualization Panel within VALSE may be similar to how a user interacts within Rhinoceros. Drawing elements on the floor plan to identify objcts and selecting trails like a user would select an object in Rhino are valuable interaction methods that will translate easily between systems. The layers sorting method here also works well within VALSE since the user will be sorting through a large amount of trail, object, and event data generated by the system. Rhino also has the ability to collapse either sidebar, giving the main drawing area more screen real estate with which to interact in.

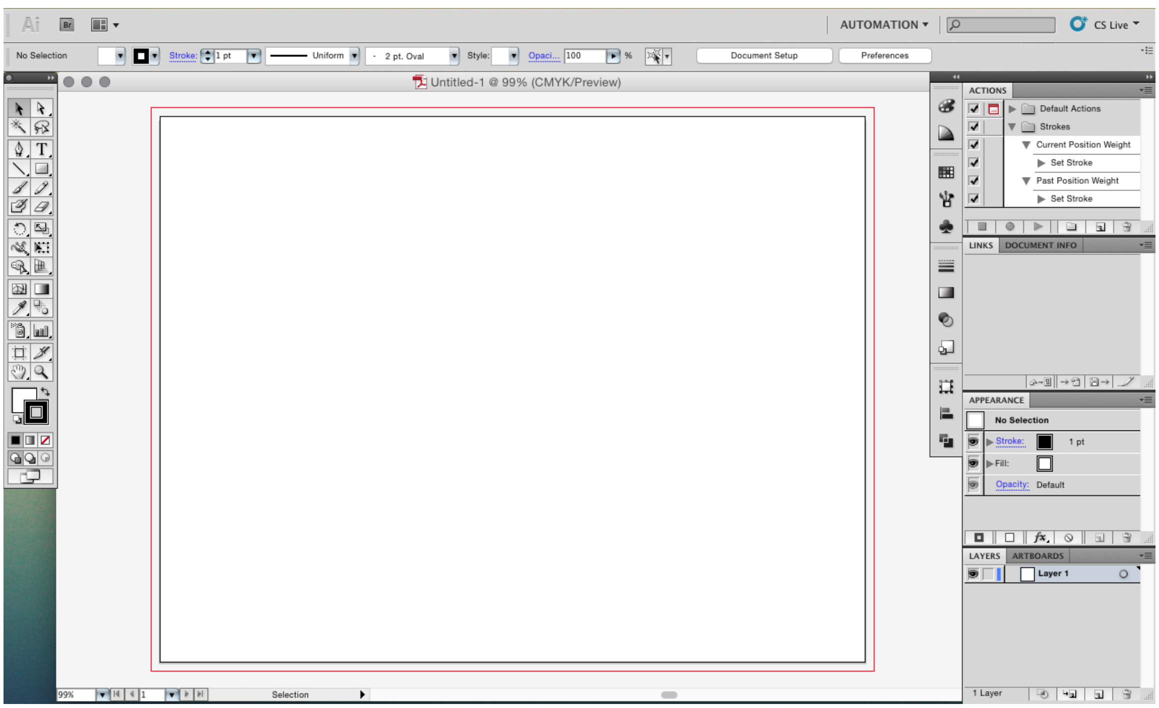
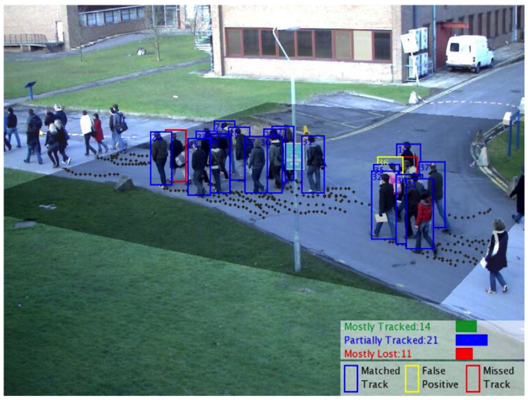
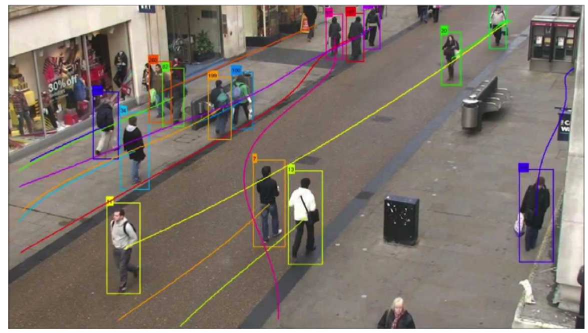


FIGURE 12: Adobe Illustrator interface.

Adobe Illustrator is similar in layout and design to Rhinoceros. It is a drawing software that utilizes similar actions of selecting objects drawn on the document as well as sorting these objects through layer control. Just like before, these means of interaction will translate into object and pathway control, sorting, and identification within VALSE.

In addition to the interface design precedents, I also looked at other representation precedents to model the data gathered by VALSE. These precedents address means of representing trail data as well as heat map data.





(Left) FIGURE13: Dot trail pathway representation. (Right) FIGURE 14: Vector trail pathway representation.

A persons trail within VALSE can be represented in one of two ways: Dot trails, or Vector trails. In the Dot trail method, the people are tracked by adding a marker every time interval. We can use the intervals to assess whether or not an individual is speeding up or slowing down based on the space between dots. This method does not show direction of travel without the context of the individual, and can get confusing if there are many trails overlapping. This is especially so in Figure 13 where color is not used to differentiate the dot trails.

Vector trails, like the dots, can become extremely jumbled and confusing if too many are view simultaneously, but are easier to distinguish from each other because of their linear nature. Like dots, vectors do not show directionality without context. Figure 14 represents vector ant trail modeling. Each person is outlined in a bounding box, and their pathway is represented by the travel vector line. The lines correspond to the boxes in color, making the paths and their creators easier to distinguish from each other. This visualization still needs the context to provide clarity to the visualization. In addition to the different trail representations, the VALSE system also uses heat maps to show aggregate data about an environment.

FIGURE 15: Ekahau interface.

The Ekahau system tracks people within a space based on RFID tags and badges that they wear. Figure 15 illustrates the way in which a user can view the relative strength of the receivers installed within the building. Greener places on the map illustrate a stronger signal and yellow spots show a weaker signal strength. This same style of representation can be applied to VALSE within the heat map section of representing the occupation data. Warmer or more concentrated hues may represent a higher population over time or larger concentration of events.

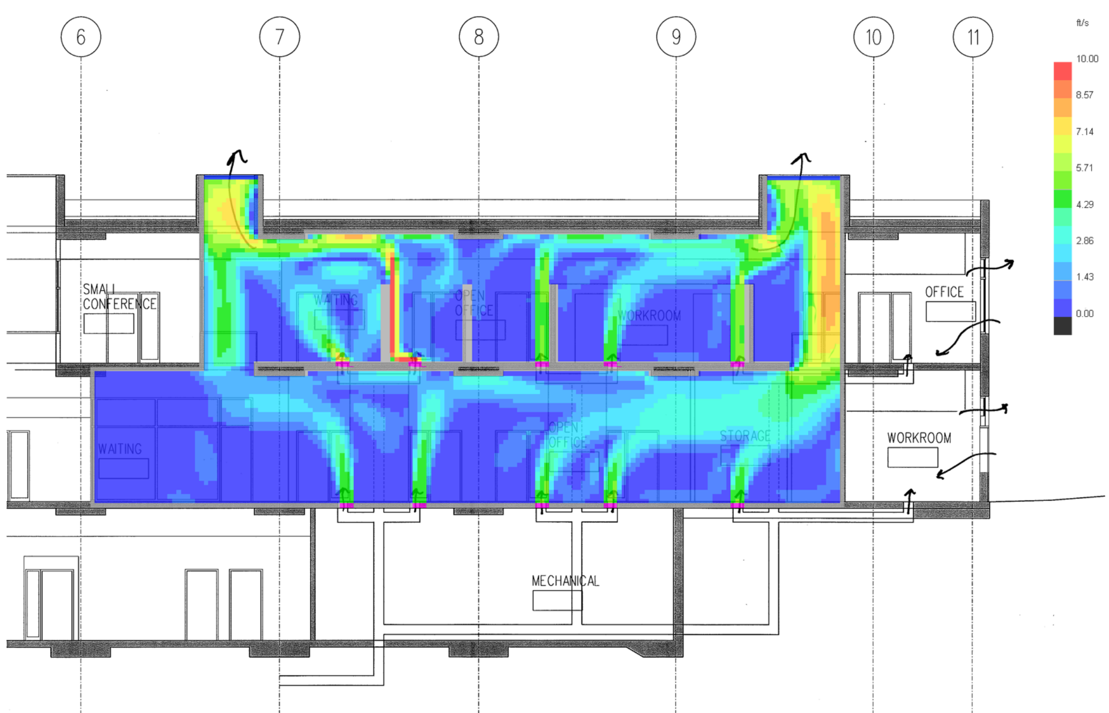


FIGURE 16: CFD Simulation heat map.

Figure 16 shows another type of heat map that is not based on the number of occupants in a space or the number of occurring events, but is based on the speed of particles moving within the space. The above figure shows high speed air as represented by warm colors, and slow speed air as cool. VALSE can apply these same principles to the way that people move through the space. This method could identify areas within the observed space that form bottlenecks or help user identify thoroughfares and routes people take in a space.

The last interface and controller example is that found within calendar software such as iCal or Outlook. VALSE can apply the same logic that these programs use when creating repeating events to analyzing data on a repeating timescale. This allows the user to set regular, repeating times, or become more specific with narrowing down which days and times to view.

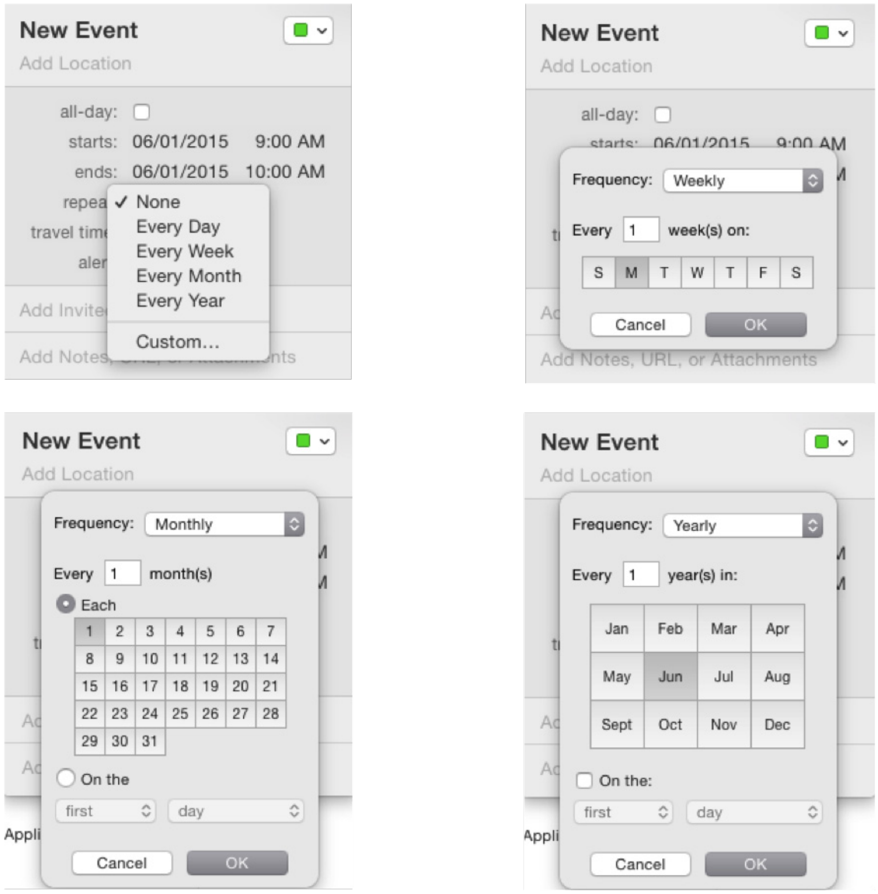


FIGURE 17: iCal event creation interface. This interface describes how a user can create a repeating event within iCal. VALSE can use these same principles in it's timeline creation method.

CHAPTER 4: VALSE INTERFACE DESIGN

As a whole, the VALSE system was modeled off of a combination of two different types of tools: drawing and annotation software. The drawing software (Adobe Illustrator, Rhinoceros, and AutoCAD) helped to lend methods for organization of elements as well as creation and selection processes. Elements such as layers and the methods of selecting and interacting with the elements in the Interaction Window were drawn heavily from the ways in which a user would interact with similar elements in drawing software. Likewise, the annotation software (ELAN annotation, VCode, and Morae) lent itself to the control of the timeline and recording the events identified by both the user and the system. We modeled the functionality of VALSE from the same mechanics presented to us in these systems so that a certain amount of familiarity would be maintained when operating an annotation system that deals with the added dimension of space to that of time.

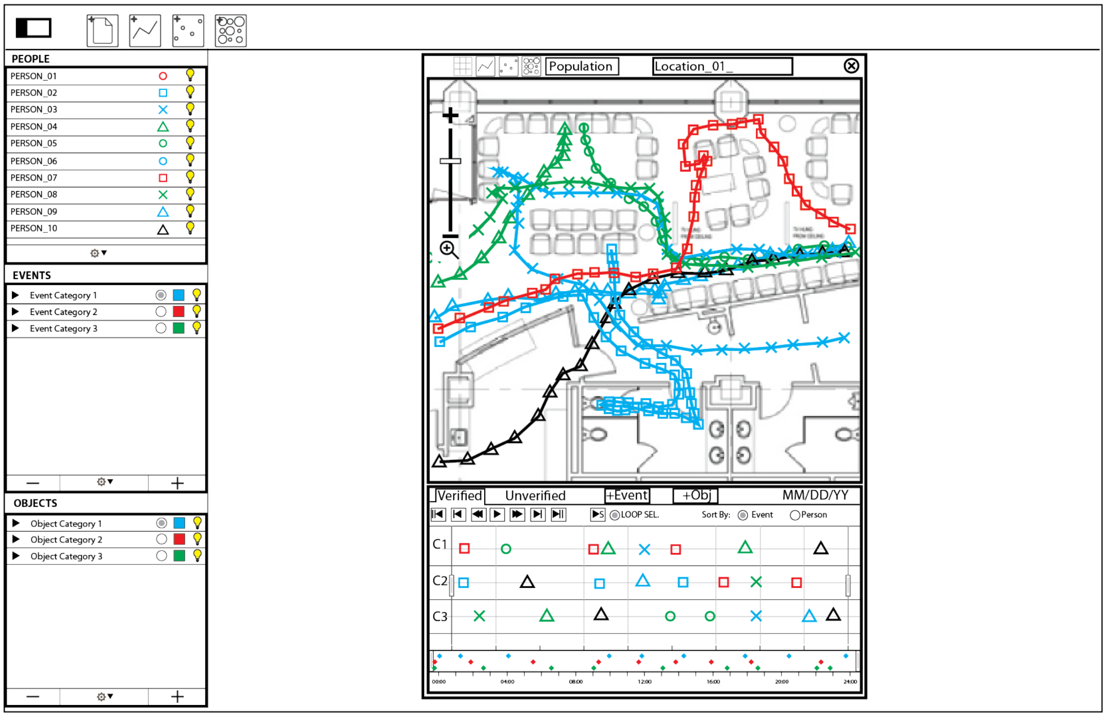
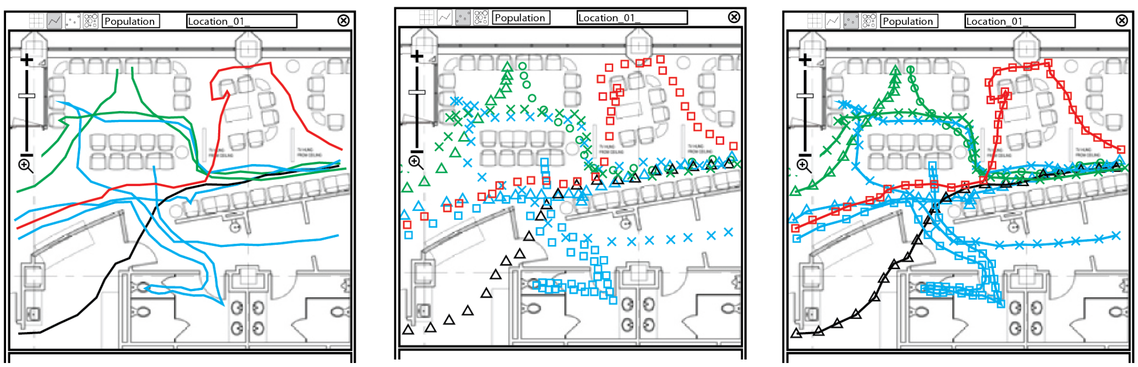


FIGURE 18: VALSE wireframe interface. The left side of the interface shows the layers sidebar. The right side window is the Interaction Window that is divided into the Visualization Panel at the top and the Event Timeline at the bottom.

The user interacts with VALSE through the Interaction Window. The window is broken into two main portions: the Visualization Panel and the Event Timeline. The Visualization Panel is the region that displays spatial information about the observed area, and the Event Timeline shows and allows the user to interact with a list of the events that have occurred within the space.

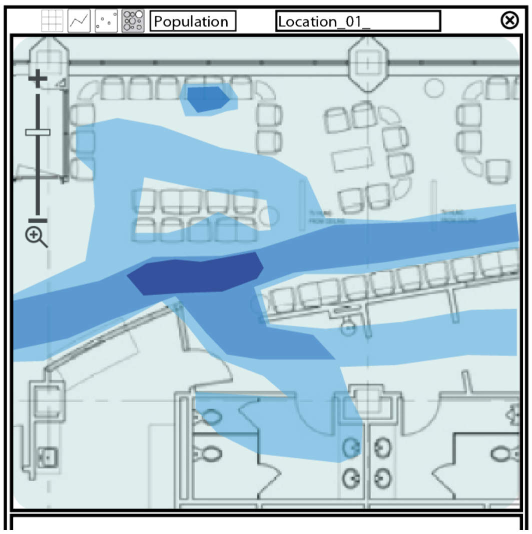
4.1. Visualization Panel

The Visualization Panel shows the user what is happening within the observed space. The image displayed on the Visualization Panel is either a floor plan view of the space with composite data gathered from the cameras capturing the movement within the observed area, or a series of snapshots taken by those same cameras.



(Left to Right) FIGURE 19: Vector trails. FIGURE 20: Dot symbol trails. FIGURE 21: Combination dot and vector trails

The pathways of the people moving through the space can be displayed in multiple ways chosen by the user. The default display for these paths is the dot trail. This type of representation shows the path of a person as a series of symbols plotted on the floor plan. These symbols are colored “x’s”, circles, squares, or triangles and can be changed by the user to help differentiate between multiple people in the space. As VALSE tracks the people in the space, it places a dot marker at the person’s location at regular intervals. By doing so, a user can infer the speed of the people moving through the region based off of the spacing of the dots. The closer the dots are to each other, the slower the person is traveling through the space. The downside to dot trails is that they can become convoluted and confusing. This is especially so if the person doubles back on their path or moves about in a small area. To provide clarity to that path issue, we included the option to view these trails as lines as well. This representation shows the user how the person moved through these confusing moments, but does not give the user an idea of direction or speed. Both of these trail display options may be toggled on or off depending on the users preference. They may also be toggled on simultaneously.



(Left) FIGURE 22: Population heat map. (Right) FIGURE 23: Snapshot view.

In addition to the dot trails, heat maps are another option for the user to display space occupation information. These maps display the aggregated data of the population in the space over the selected time period. This data may be shown to the user as population density, travel velocity, or event hotspots. For the population density heat map, the warmer colored areas represent a higher number of people occupying the designated space. This type of display shows the user areas where people tend to move through or gather and can be useful for looking at a large set of trails in a cleaner manner than overlaid dot or line trails. The velocity heat map uses these same principles, except that it maps areas of high average speed with the warmer colors. Users can use this heat map to identify bottlenecks or thoroughfares within the space. Finally, the event hotspots heat map shows the user the frequency of events within the space over the selected time. The more events that occur in a specific area, the warmer the color will be. This can help the user identify regions where there is a high level of notable activity.

In addition to the two abstracted views presented in the Visualization Panel, there is a third option that shows snapshots taken by the cameras of the selected location. The Snapshot view divides the panel into a series of smaller panels, each containing a snapshot from a single camera. These images show the state of the environment at the time marker on the Event Timeline. Each of the smaller image panels can be minimized or maximized within the Visualization Panel to allow the user to see more or less detail about the space. These images provide the user with more context than an abstracted trail or heat map could for a single event, and will help the user identify events on the timeline and label them correctly.

4.2. Event Timeline

The Event Timeline allows the user to navigate the information presented to the user from the Visualization Panel. It is divided into two main sections: the control toolbar and the timeline panel.

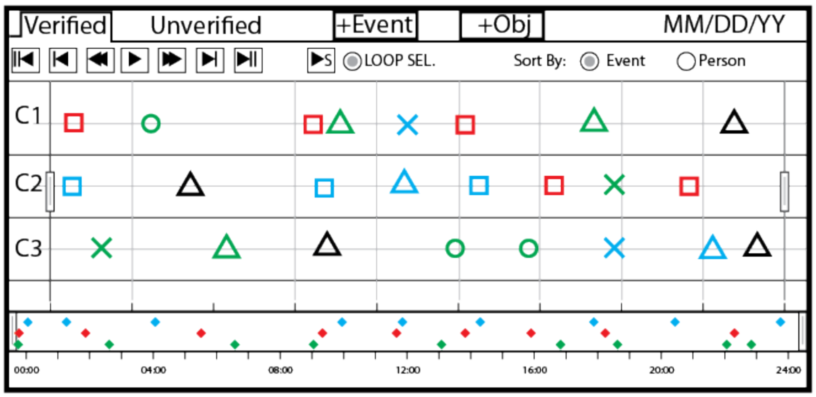


FIGURE 24: Event timeline.

The Control Toolbar portion of the Event Timeline allows the user to make changes to the selected time as well as playback the recorded information. The Datetime function on the toolbar allows the user to specify a timeframe for the system to analyze. The operation of this function was drawn from event or reminder creation in calendar software. It allows the user to examine single moments or recurring time spans within the recorded data set. For example, a user may wish for the Interaction Panel to show information about the space for every Monday from 10:00-11:00am. Likewise, a user could define a single timespan that displays data from the space for a few hours, days, weeks, etc. The toolbar also includes a set of playback controls that allows the user to navigate through the selected time period at a variety of speeds both forward and backwards. In addition to these time control buttons, the New Event button is also located on this toolbar, but will be addressed further into this document.

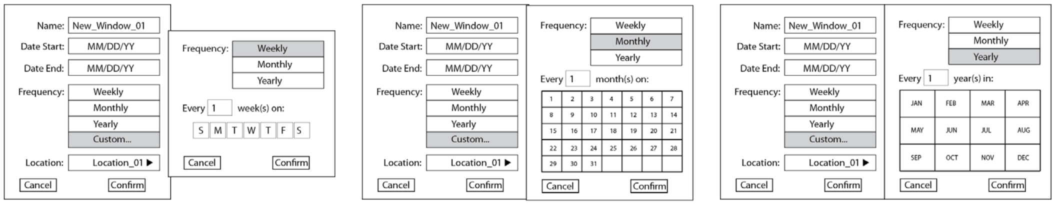


FIGURE 25: Datetime menu.

The timeline panel shows the user when events that are identified by either the system or a user occur. The timeline has two sorting options for displaying the events. The first is sorting by person. This method sorts the list of people in the space vertically and displays when an event occurs by placing a colored marker at the appropriate point in the timeline. In this way, a user may see all of the events performed by a specific person in the space regardless of what category the event falls under. In this sorting option, events are shown as either simple or compound. Simple events are those that occur without the influence of other events performed by the person in the space. An example of a simple event would be a person passing through the space without stopping. It is a single event made of one component. Compound events are made of multiple simple events chained together, much like a compound sentence is made of smaller sentences chained with conjunctions. An example of a compound event is “waiting”:

1. A person enters the space (in this case it is a waiting area) and checks in with the receptionist. This is a simple event that may be labeled as “checking in”.
2. The person then goes to a chair and sits down. This is the simple event, “sitting”.
3. The person stands up and moves to the magazine rack to pick up something to read. This is the simple event, “browse magazines”.
4. The person then returns to his/her seat. Once again performing the event “sitting”.

Here, “waiting” is composed of four distinct, simple events all performed by the same person in succession. These events would still be marked on the timeline with their appropriate category color, but would also be joined with a line denoting that they are a part of a compound event.

The second option for sorting the timeline is by event. Rather than being sorted vertically by person, the timeline is sorted vertically by event category. The timeline now uses markers to denote who preformed the event. They are marked similarly to the previous sorting method, but are marked with the colored symbols used to identify the person performing the event. Representing events in this manner only displays the events as simple events. A user may view the events on the timeline in this manner to gain an understanding of how frequently a specific event occurs or when these events occur in relation to each other.

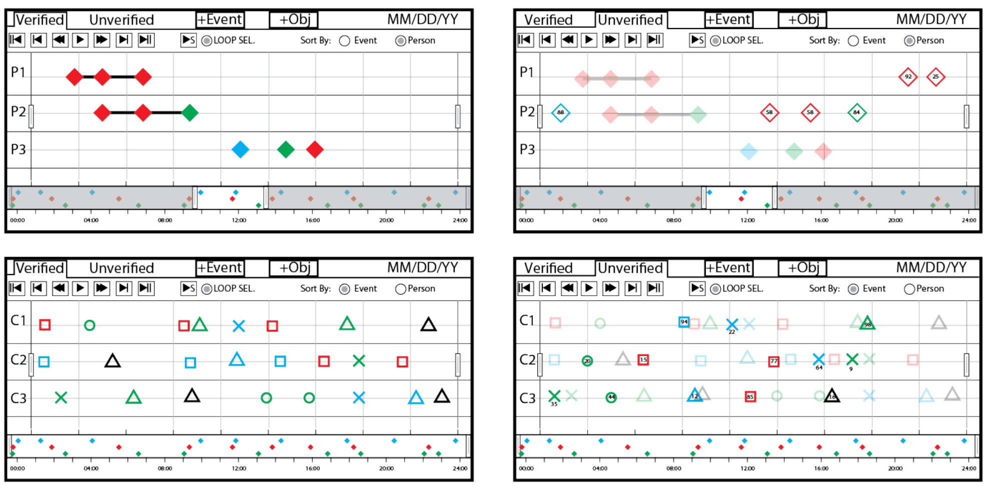
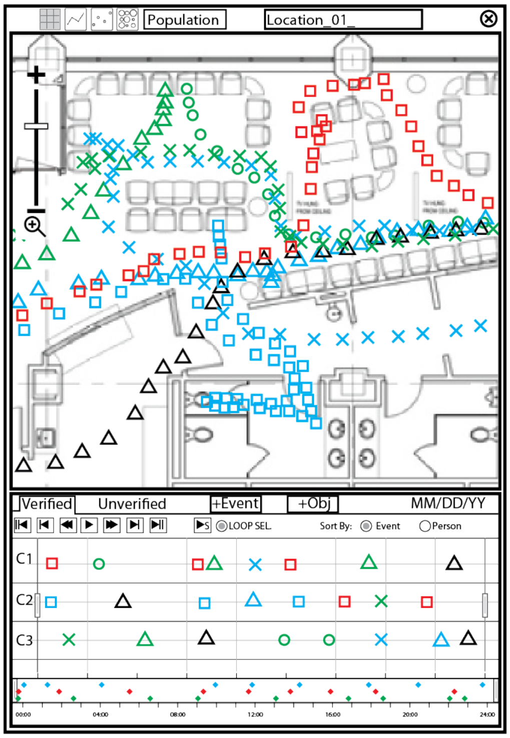
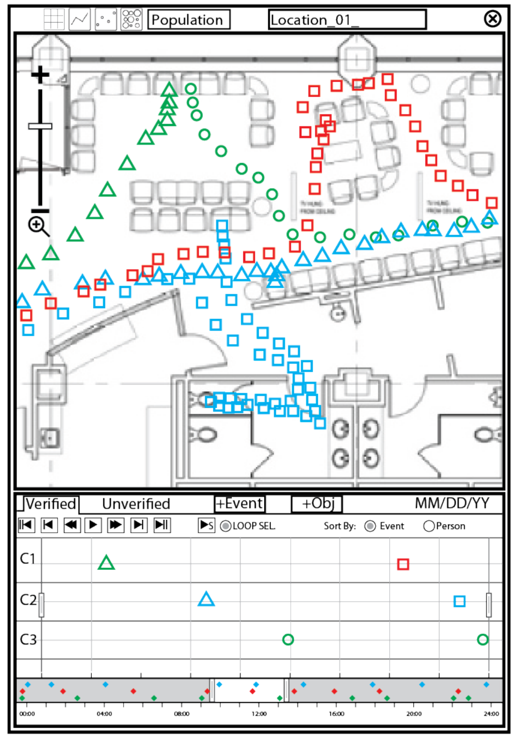


FIGURE 26: Event timeline examples.

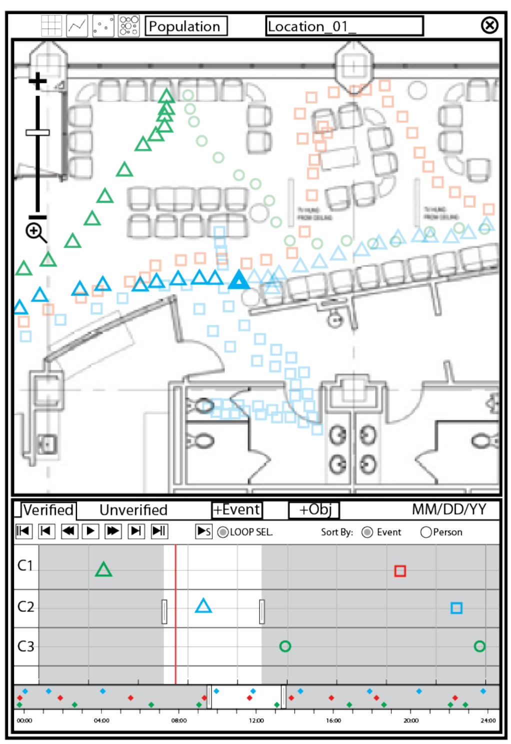
In addition to these two views of the events on the timeline, there are two tabs that allow the user to switch between Verified Events and Unverified Events. The Verified Events tab shows the user all of the manually generated events and the system generated events that the user has approved. These events are represented as filled in diamonds. Unverified system generated events are shown as open diamonds with a confidence rating inside. This rating shows the user how confident the system is about the accuracy of the identified event. Once a user verifies an event, it is moved to the Verified Events tab. Otherwise, the unverified event will remain until the user deletes it.

The last part of the Timeline Panel is the Overall Timeline. This is the bar at the very bottom of the Event Timeline that shows the user an overview of the events during the time selected. This bar allows the user to control what is being displayed in the above Timeline Panel. There are two handles on either end of the Overall Timeline that the user can move to identify the beginning and end time for the observed area within the selected timeframe. Moving these handles does not change the Overall Timeline, but does change the Timeline Panel to show only the events between the handles. This action also changes the data being shown in the Visualization Panel. The trails of those moving through the space during the focused time become the only visible trails, and the heat maps only calculate what to display based on those paths. This method of using handles to focus the view is also implemented on the Timeline Panel. The same style of handles can be used to get an even finer focus on the desired information. Performing this focus changes the Visualization Panel to display the data gathered between the handles on the Timeline Panel. The dot trail representation also changes slightly when this level of focus is desired. Rather than showing the trails as the same weight, as the system did without the Timeline Panel focus, the trails now show three different weights according to the position of the time marker. The dots that represent the position of the person at the time marker are identified as bold symbols. The previous dots are marked by medium weight symbols, and future dots are identified by lightweight symbols. This representation gives the user more context about the people in the environment, and helps the user determine which direction the people are moving within the space.





(Left) FIGURE 27: Unfocused timeline.
(Right) FIGURE 28: Focused overall timeline.
(Below) FIGURE 29: Focused timeline panel.



4.3. Event Creation

A user creates an event by clicking on the New Event button located in the Event Timeline Control Toolbar. This prompts the user to select the people involved in the event as well as any associated objects. As a rule, there is always at least one person involved in an event, but there may be no objects involved. The elements may be selected on the Visualization Panel or on the Layers Sidebar. After the user has selected the elements involved in the event, the next step is to select the beginning and end of the event. This can be accomplished in one of two ways. The first is to move the handles in the Timeline Panel to encompass the duration of the event. The second is to select the dots of the trails in the Visualization Panel. The user first selects the dot where the action begins, then selects the dot after the event is completed. Using this method, the user can utilize spatial cues to help identify when the event begins and ends. Utilizing machine learning techniques, manually identifying events will teach the system what to look for when generating system-defined events. The more manually generated events and verified system-generated events there are in the database, the more accurate the system will be at identifying events on its own.

Along side of the top-down approach of teaching the system by example through machine learning, we are also looking to approach the design of VALSE from a bottom-up method as well. To do this, we've formulated a series of "recipes" that utilize the principles of spatial proxemics.

Spatial Proxemics in Event Identification:

Intimate Space: This space can be used to define if a person is directly interacting with an object or person. For example: sitting in a chair, interacting with a desktop, washing hands, opening a door, hugging or kissing another person.

Personal Space: This space can be used to define if a person is interacting with another person on a more colloquial level or observing something from a short distance.

Examples: A person reading a bulletin board, a couple holding hands, a handshake greeting, or a hushed conversation.

Social Space: This space may be used to define if two people are interacting with each other on a less colloquial level, or if a person is looking at something from afar. Examples: Two people walking or talking with each other, two people interacting over an object such as a reception desk. This could become an issue in the case of a crowded area. There could be many false positives for this type of interaction. This could be solved with a "future studies" set to develop a "following" or "togetherness" metric

Public Space: Maintaining a public space distance around a person for the duration they are in a space may indicate that the person had no intentions of using anything in that space or interacting with anyone. This could be used to identify a person moving directly through a space.

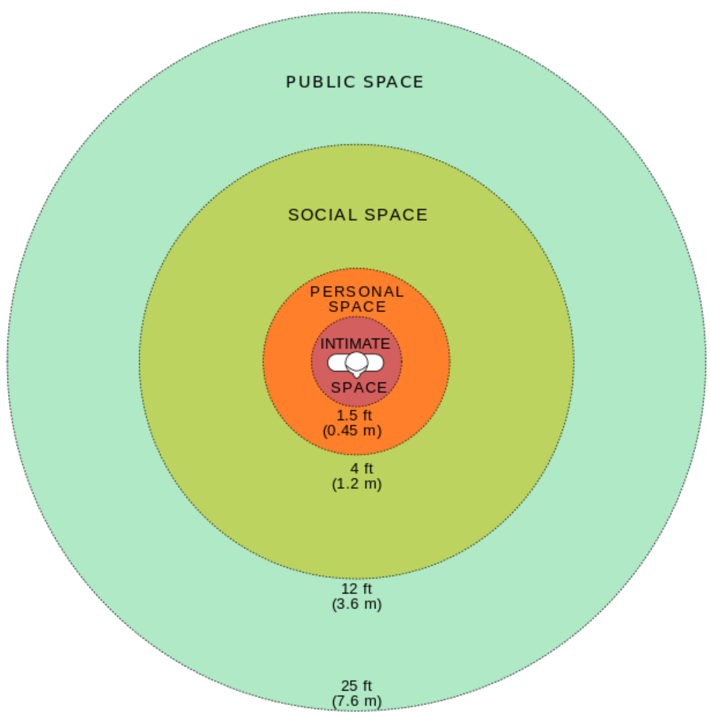


FIGURE 30: Spatial proxemics radii.

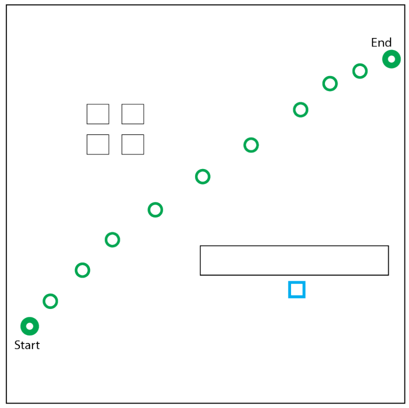


FIGURE 31: Simple event - Passing through.

Person (simple event)

This type of event involves only one person and is contingent on the fact that this person spends no significant amount of time interacting with objects or other people in the space. For example, a person passing through a space without stopping could be categorized as “passing through.”

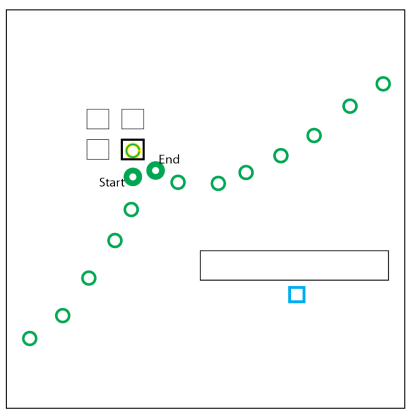


FIGURE 32: Simple event - Sitting

Person – Object (simple event)

These types of event involve a single persons interaction with an object over the course of some amount of time. For example, a person enters the area that defines a chair for an extended period of time. This can be defined as a “sitting” event.

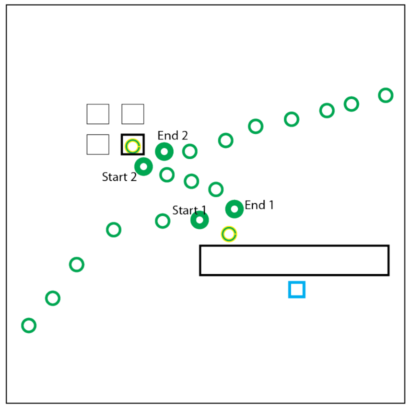


FIGURE 33: Compound event - Checking in and waiting.

Person – Object + (compound event)

These types of events involve a single person interacting with two or more objects in series to define an event. For example, a person enters the area defining a receptionist desk for a period of time and then enters the area defining a chair for an extended period of time. This could be defined as a “checking in and waiting” event.

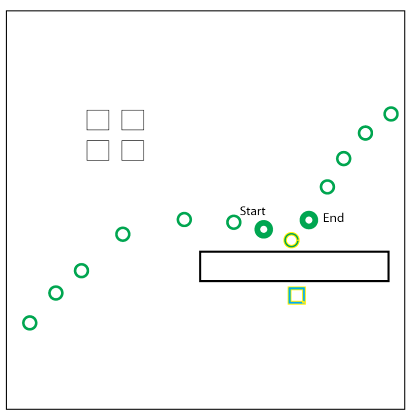


FIGURE 34: Simple event - Conversation

Person+ – Object (simple event)

These types of events involve two or more people and a single object. This type of interaction could define a specific area where interactions may occur on a fairly consistent basis. For example, two people are within a certain distance of each other with an object defined as the receptionist desk between them. This could be identified as “conversation” or “question”. Alternatively, this could also be used to identify when multiple people are working together with a piece of equipment.

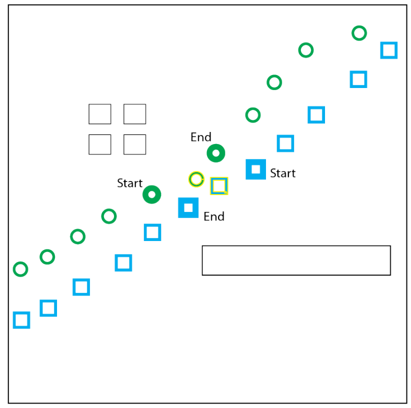


FIGURE 35: Simple event - Meeting

Person – Person+ (simple event or compound event)

This type of event involves two or more people interacting in an undefined region of the observed space. A simple event example is a group of people converges at a spot within the observed area for an extended period of time. This event could be recorded as “conversation” or “meeting” or “chance encounter.” A compound example may be a person goes from person to person in a room asking them for something or giving something out.

All of the previous examples can be combined or multiplied to form compound events.

Part of the system design with VALSE is that the system learns about events as they are created and verified by the user. As a result of this, one of the design goals for the system is to allow VALSE to look at past data and retroactively generate events based on new data fed to the system. Patterns of action recognition may develop over time and fit past data points.

4.4. Element Management

As mentioned previously, one of the ways to help identify where events occur in space is through people's relationship to fixed objects within the space. An object is a user-defined area that labels a stationary object within the observed area. These may be actual objects such as a chair or a desk, but they may also be areas of interest such as a room's threshold. Objects are persistent elements and appear in every new instance of the area in which they were initially created. Objects are created using the "create object" button in the top toolbar of the Interaction Window. Clicking this button allows the user to identify the area of interest with a pen-tool similar to that in Adobe Illustrator. The pen allows the user to define the boundaries for the object and can accommodate various different shapes that are appropriate to the location. Once created, the user or system may then include the new object in any event creation, retroactively or proactively.

The Layers Sidebar gives the user another way of interacting with the system apart from the Interaction Window. Selections made within the layers sidebar are displayed across all platforms of interaction. For example, selecting an event from the Events sidebar will highlight the corresponding event within the Event Timeline, any object(s) associated with the event in both the Visualization Panel and Objects Sidebar, and any person(s) involved in the event both in the sidebar and in the Interaction Window. The sidebar performs much the same as the layers window within Adobe Illustrator or AutoCAD. Colors and symbols associated with identifying People, Events, and Objects can all be changed. Visibility can be toggled on and off and elements can be renamed. Additionally, the Layers Sidebar can be collapsed to give the user more screen real estate for the Visualization Window(s).

Some actions and attributes that are specific to the People Sidebar are as follows: people cannot be created or deleted and multiple people can be merged into a single person or broken apart if desired. Since people are elements that are identified by the cameras capturing the space, people cannot be added or removed by the user. However, people can be joined into a group that represents a single person. If the user observes that the same person leaves the observed area and reenters it, he or she may merge the two trails created by the system into a single entity. Since the computer cannot determine if two people occupying the observed space at different times are the same person, only a user may make this merge. This action cannot be system-generated. Merging two trails together also merges the events associated with each trail onto the same person, so clicking on the merged person will display all events that were associated with the all of the component people.

FIGURE 36: People layer sidebar options.

Events and Objects in the sidebar are organized according to category. For example, there may be a "sitting" category that contains all of the events of people sitting within the observed space or a "chair" category holding all of the objects identified as chairs. When a new event or object is created, the user has the ability to choose which category it will be located under, but the currently selected event will be the default. Both types of elements can be moved and copied from category to category.

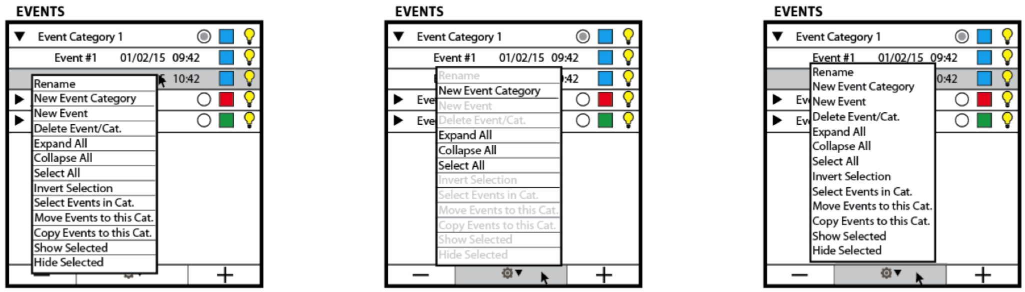


FIGURE 37: Event layer sidebar options.

4.5. New and Multiple Interaction Windows

Upon opening the system, the user is prompted with an initial dialogue box to create a new Interaction Window. Here, the user defines which location he or she wishes to observe and the timeframe to apply to that location. VALSE then generates the proper Visualization Window with all of the events, people, and objects displayed. The defaults for creating a new Visualization Window are based on the most recent data gathered by the system. The default location is the location that was last used by the user, and the default time is the past twenty-four hours.

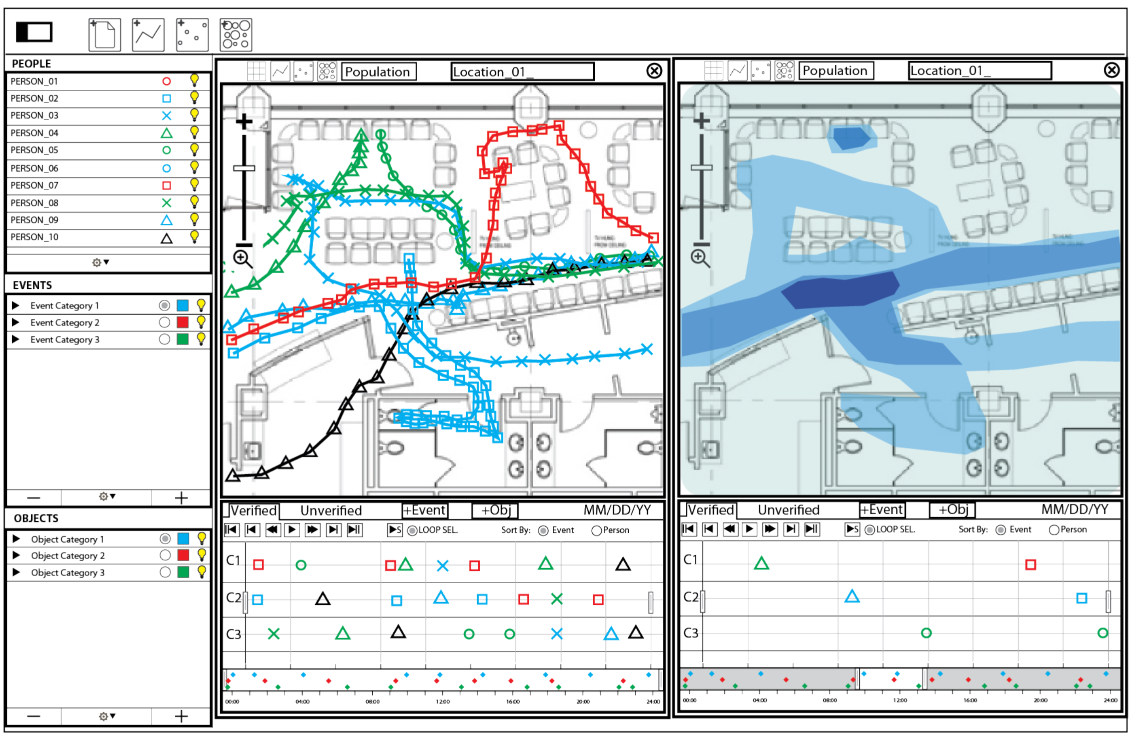


FIGURE 38: Two interaction windows.

One of the features of VALSE is that a user may create more than one Visualization Window for comparison purposes. Pressing the "new Interaction Window" button in the top toolbar creates these windows. It prompts the user the same way the system does upon startup, and a second window is created alongside of the existing one. When two or more Interaction Windows are open within the system, a "Master Playback Control Bar" pops up from the bottom. This playback bar functions the same was as the one within the Event Timeline, except that this toolbar controls the playback for all Interaction Windows open within VALSE. This way, a user can synchronize the playback of multiple time-lines.

Similar to creating a new Interaction Window, there is a series of "new visualization" buttons in the top toolbar of VALSE. Each button represents a different visualization available within the system, and clicking one will add a second visualization to an existing Interaction Window. If there is only one Interaction Window open, the system will simply add the selected visualization to that window. If there are more than one open, the system prompts the user to select the desired Interaction Window to apply the change to. The resulting Interaction Window is composed of two Visualization Panels and a single Event Timeline that controls both visualizations.

Two Interaction Windows can be merged resulting in a single Interaction Window with two Visualization Panels and one Event Timeline controlling both of them. To do this, a user must open at least two Interaction Windows, and both must be displaying the same observation space. Then, the user may drag one window on top of the other, prompting the user with a dialogue window. Here, the user may choose to:

1. Use the timeline from the first window with the two visualizations.
2. Use the timeline from the second window with the two visualizations.
3. Use the join of both timelines with the two windows.

Options 1 and 2 work the same way as adding an additional visualization to an existing Interaction Window. Option 3 works similarly, but in addition to adding a second visualization, the timelines join as well. For example, Timeline_01 is examining every Monday in January from 10:00-11:00am, and Timeline_02 is examining Thursday, January 19th from 12:00am-11:59pm. The join of these two timelines would be a single timeline showing each Monday from 10:00-11:00am with the Thursday time inserted at the appropriate point. The advantage of this is that the user can examine aggregate data for complicated time structures.

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

The ideas and methodologies presented here for developing a VALSE user interface have led the lab to begin creating a Java based interface for the project. This interface has also been shown to multiple healthcare architecture firms through its development, and has begun a research collaborative between one of these firms and the School of Archtiecture at UNCC. The future development of this system will entail more detailed refining and adding elements to the wireframe as functionalities dictated by both architecture firms and ethnographic end users manifest themselves.

As architects, VALSE would be able to give us insights as to whether that curved wall did actually funnel circulation in the desired direction. Or if placing a staircase in that certain place really did improve the function of the space. I mentioned healthcare specifically earlier in this paper, but VALSE is not limited to only that field. VALSE can provide services to many different buildings that have a repeatable typology: Educational facilities, retail, and office spaces are only a few. By embracing technology and its incorporation into new architecture, we can design smart buildings that will tell us more about what is going on inside of them than ever before.

BIBLIOGRAPHY

Akdemir, U., P. Turaga, et al. (2008). An ontology based approach for activity recognition from video. 16th ACM international conference on Multimedia ACM.

Badram, Jakob E., et al. (2006) Experiences from Real-World Deployment of Context Aware Technologies in a Hospital Environment. Ubicomp 2006 pp. 369-386 2006.

Beorkrem, C. and E. Sauda (2008). Architecture User Interface: Towards a Performative Architecture. 2008 EAAE/ARCC Conference: Architectural Research and the Digital World, Copenhagen.

Camurri, A., P. Coletta, et al. (2004). Toward real-time multimodal processing: EyesWeb 4.0.

Chan, A. B., Z. S. J. Liang, et al. (2008). Privacy preserving crowd monitoring: Count ing people without people models or tracking. Computer Vision and Pat tern Recognition, IEEE Conference on.

Chen et al. (2011) Discovering social interactions in real work environments, Automatic Face & Gesture Recognition and Workshops (FG 2011)

Cioffi-Revilla, Claudio (2010) Computational Social Science, WILEY Interdisciplinary Reviews: Computational Statistics, Vol. 2 no. 3, May/june 2010 pp 259-271

Dai et al. (2009) Group Interaction Analysis in Dynamic Context. Systems,

Dourish, P. (2006). Re-space-ing place: place and space ten years on. Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work: 299-308.

Forsythe, D. E. (1999). ““It’s Just a Matter of Common Sense”: Ethnography as Invisible Work.” Computer Supported Cooperative Work (CSCW) 8(1): 127-145.

Gavrila, D. M. (1999). “The Visual Analysis of Human Movement: A Survey.” Computer Vision and Image Understanding: CVIU 73(1): 82-98.

Ge, W., R. T. Collins, et al. (2009). Automatically detecting the small group structure of a crowd. Applications of Computer Vision (WACV), 2009 Workshop on: 1-8.

Groat, Linda, David Wang. Architectural Research Methods, 2nd Edition, Chapters 7-8 pages 215-312. Wiley 2013

Ivanov, Y., C. Wren, et al. (2007). "Visualizing the History of Living Spaces." IEEE Transactions on Visualization and Computer Graphics 13(6): 1153-1160.

Life in the network: the coming age of computational social science: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2745217/

Low, Setha. Towards an anthropological theory of space and place. Cit University of New York, Walter de Gruyter. 2009.

Raptis, M. and S. Soatto (2010). "Tracklet descriptors for action modeling and video analysis." Computer Vision-ECCV 2010: 577-590.

Rosenstein, B. (2002). Video use in social science research and program evaluation. In ternational Journal of Qualitative Methods 1 (3). Article 2.

Ruback et al. (2011) People Transitioning Across Places: A Multimethod Investigation of How People Go to Football Games. Environment and Behavior 2011

Spurlock, S. and R. Souvenir (2012). Dynamic subset selection for multi-camera track ing. Proceedings of the 50th Annual Southeast Regional Conference, ACM.

Yu et al. (2009) Monitoring, recognizing and discovering social networks, CVPR 2009

Zhang et al. (2004) Modeling individual and group actions in meetings: a two-layer hmm framework, CVPR 2004