Viz-A-Vis: Toward Visualizing Video through Computer Vision

Mario Romero, Jay Summet, John Stasko, *Member, IEEE*, and Gregory Abowd, *Member, IEEE*

Abstract— In the established procedural model of information visualization, the first operation is to transform raw data into data tables [1]. The transforms typically include abstractions that aggregate and segment relevant data and are usually defined by a human, user or programmer. The theme of this paper is that for video, data transforms should be supported by low level computer vision. High level reasoning still resides in the human analyst, while part of the low level perception is handled by the computer. To illustrate this approach, we present Viz-A-Vis, an overhead video capture and access system for activity analysis in natural settings over variable periods of time. Overhead video provides rich opportunities for long-term behavioral and occupancy analysis, but it poses considerable challenges. We present initial steps addressing two challenges. First, overhead video generates overwhelmingly large volumes of video impractical to analyze manually. Second, automatic video analysis remains an open problem for computer vision.

Index Terms— Spatiotemporal visualization, time series data, video visualization, sensor analytics, image/video analytics.

1 INTRODUCTION

Many disciplines spend considerable resources studying activity and behavior [2-4]. Methods range from qualitative pen-and-paper observation [5] to automatic video content analysis [6]. A method's appropriateness depends on the analytical goal, the observable features of target behaviors, the observers' tolerance to ambiguity, and the subjects' tolerance to intrusiveness. We present a method that is appropriate for variable term, continuous and high-resolution analysis of subjects that consent to overhead camera observation.

Overhead video has the temporal and spatial resolution to potentially open new insights into everyday human behavior by objectively revealing its invisible spatiotemporal structures, large [7] and small [8]. If analyzed thoroughly enough, it may function as a window into how people relate to each other and how they appropriate natural spaces and the objects within. Overhead video has potential for new analytical applications in multiple areas. For example, it may be applied to the long-term objective evaluation of behavioral therapy in special classrooms. It may track developmental progress in a baby's nursery. It may provide objective, long-term, and continuous physical therapy progress reports in natural environments beyond the doctor's office. It may quantify detailed occupancy for the analysis of architectural design, trace factory operations to increase industrial productivity, and discover customer behaviors to boost retail space marketability.

While overhead video presents abundant analytic opportunities, it also introduces important challenges. First, it rapidly generates overwhelmingly large data sets for manual analysis. Second, reliable high level automatic analysis remains elusive. Third, video intrudes on privacy. We address the first two challenges.

We present Viz-A-Vis (Figure 1), an overhead video capture and access system [9], as an initial approach to building information visualization interfaces on top of computer vision abstractions. Our focus is on bridging the semantic gap between high level human analysis and low level machine sensing [10]. From the computer's side, we bridge the gap through simple but robust computer vision. From the human's side, we bridge it with information visualization methods. Bridging the gap with computer vision alone remains an open problem. Bridging it with visualization alone requires significant user input and is impractical for lengthy video.

An important difference between computer vision and information visualization is the source of inference. In computer vision, inference occurs in the machine. In information visualization, it is centered in the user's cognitive and perceptual structures. We explore the rich potential for a mixed-initiative interface [13]. We use simple low level computer vision to hide most of the unnecessary detail in raw video, but purposely avoid higher level abstractions that introduce brittleness. Illustrated in Figure 2, our model of video visualization keeps the human at the core of inference and places computer vision as a support structure for data transformation.

We have explored several methods to tackle the privacy challenge. We have continuously deleted original frames [11] and saved only relevant processed frames that eliminate identity. We have experimented with physical blur filters [12]. We have given users control to stop and delete data capture [13]. While these techniques have a detrimental effect on potential analytic depth due to lower raw data quality, they still depend on subjects' trust. Privacy remains an open concern for all sensing technologies. Ultimately, it is up to subjects to consent to the sensing and to trust that the data will be ethically handled. Subjects should decide if benefits outweigh costs. This is the principle of proportionality [14].

In the following section we contextualize Viz-A-Vis within related work. In section 3, we describe in detail the architecture of Viz-A-Vis. In section 4, we present a preliminary case study where partial use of Viz-A-Vis opened new insight into behavioral patterns. Lastly, we conclude and propose future work.

2 RELATED WORK

Viz-A-Vis is a multi-disciplinary tool. It employs numerous methods from computer vision, information visualization, and video content analysis. We explore the relation with the most relevant work only.

Ivanov *et.al.* present a visualization of the history of living spaces [15]. They visualize multimodal, long term sensor data that include a number of motion detectors and video cameras. A low level perception technique they use for the high level visualization is path tracking of people in space and time. In relation to our paper, they provide abstract visualization and navigation tools and rapid indexing to original motion sensor and video raw data. We set similar goals, but present a number of important methodological differences. First, our video data comes from overhead cameras that have a one-to-one correspondence with architectural space. Second, our goal is to study a broader range of behaviors, more than can be inferred from simply tracking paths. For example, we distinguish sitting watching television versus reading a book.

Two important contributions to this discussion line are to explicitly embed computer vision as part of the information visualization pipeline and to directly map image space to architectural space. An important difference is the price our method

- *Mario Romero is with Georgia Tech, E-mail: mromero@cc.gatech.edu.*
- *Jay Summet is with Georgia Tech, E-mail: summetj@cc.gatech.edu.*
- *John Stasko is with Georgia Tech, E-mail: stasko@cc.gatech.edu.*
- *Gregory Abowd is with Georgia Tech, E-mail: abowd@cc.gatech.edu.*

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

1077-2626/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 13:07 from IEEE Xplore. Restrictions apply.

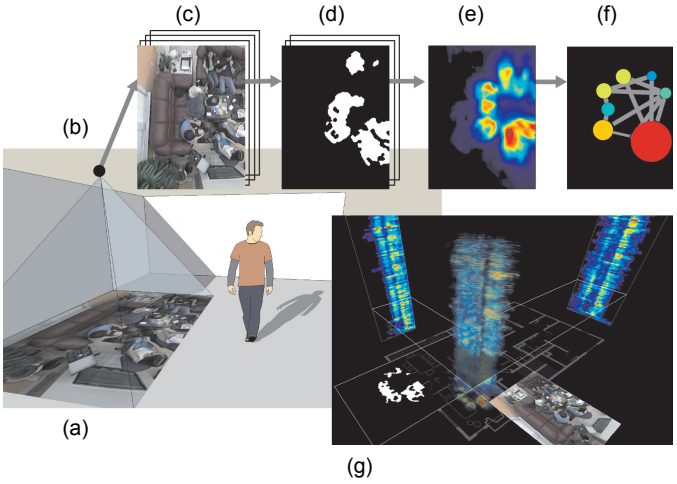

Fig. 1. Viz-A-Vis overview: capture and visualize activity. (a) Place of interest, (b) overhead camera, (c) image sequence, (d) motion sequence, (e) spatial and temporal aggregation, (f) semantic aggregation, and (g) visualization of activity.

pays for the increased data gathered and the increased cost of the sensing infrastructure. We map a pixel to approximately 1 cm^2, which means that for a large floor plan, as the one studied by Ivanov et al, we require significantly more cameras and storage space. Cameras are not only more expensive, they are significantly harder to install than motion detectors. These factors need to be included in the consideration of using overhead video. We manage the storage space problem by constantly deleting images without subject presence (zero-motion over a long enough period of time). We keep a small buffer which retroactively starts saving images once activity is detected and we continue to save images for a reasonable time after activity disappears. For all our installations (over 7500 hours), this simple mechanism reduced data gathering from about 240 to 3 GB per day, making it technically feasible to record over a month of activity with today's typical laptop internal hard-drives (120 GB). Finally, by using the third dimension of our geographical information system to map time, our method cannot directly generalize to multiple floors. Tracking multiple levels of an office building was the main reason Ivanov et al. avoided mapping time to a third dimension in their visualization.

Our general goal is to visualize a multivariate time series in its spatial context. There is a long history of proposed solutions. The most relevant to our work are [16] and [17]. Kapler and Wright contextualize time series data using the third dimension of a space-time cube that's base is the relevant 2D map. The main methodological difference in our paper is that we visualize denser data coming from overhead cameras. While GeoTime visualizes one-dimensional paths across 3D space, we visualize two dimensional surfaces. Kwan and Lee visualize large scale activity patterns in time-geographies that visualize summarized data for large populations over city-size areas. We visualize spatiotemporally dense data for small populations over house-size areas.

Video visualization is a vibrantly active field of research in recent years. Daniel and Chen present a visualization that holds many similarities to our activity cube [18]. They visualize motion in a video space-time cube. They map motion pixels to low translucency in the cube and static pixels to high translucency, thus enabling a human operator to see through inactive sub-volumes of the video cube. Other relevant approaches that model and visualize video as a space-time cube are [19-22]. Our approach takes these ideas a couple steps further. First, we directly map the video cube to a geographic information system, where the horizontal plane is both image and architectural space and the vertical plane is time. Second, we

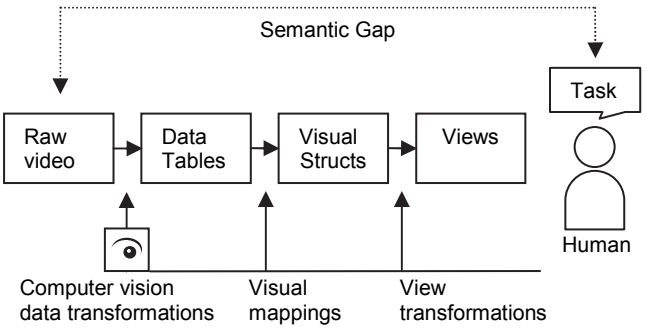

Fig. 2. Traditional information visualization procedural model [1] augmented with computer vision. Analysis remains in the human.

aggregate motion into regions of interest and linearize the aggregates into the rows of a two-dimensional matrix (the activity table) that summarizes the semantics of activity with respect to place and time.

With MUVIS Kiranyaz *et al.* present a multi-media browser with automatic low level feature extraction and high level visual summaries that support navigation, indexing, and querying [23]. The main difference with our work is that they do not contextualize their data in physical space. Their work is primarily concerned with media content and not real-world context.

3 VIZ-A-VIS ARCHITECTURE

Viz-A-Vis is a capture and access system. The capture comes from overhead cameras and the access occurs during the analytical process mediated by information visualization on top of computer vision. The video goes through two inverse processes: a process of abstraction, where relevancy is automatically detected and aggregated, and a process of reification, where visual overviews are explored, filtered, zoomed, contextualized, annotated, and indexed back to relevant video sequences. The goal is to provide a visual roadmap that serves as a video semantic navigation tool.

3.1 Process of Abstraction

The process of abstraction for sensing infrastructures begins at the selection and placement of sensors. There are usually many competing considerations, such as acuity, relevancy, and intrusiveness. The sensor should have enough acuity to capture most observable phenomena of target events. We chose cameras because they can capture most visually observable human behavior, down to single fingers moving.

The second choice is placement. We chose to place the cameras over areas of interest for several practical physical and algorithmic considerations. Physically, by being on the ceiling, cameras are relatively out of the way. Algorithmically, by having an overhead view of the world, the computation of low level vision is simplified. We have installed the system in a research laboratory, five area homes, and two museums. In each installation we carefully analyzed the space, the objects in it, and the occupancy of the space through preliminary interviews (Figure 1a-b).

In video, the process of abstraction begins at the hardware level, with quantization and discretization of time (frame rate), space (resolution), luminance (sensitivity to light), and chrominance (sensitivity to color). The camera should have the speed, resolution, and sensitivity to capture most target behaviors in its field of view for its intended application. For our applications, we used off-the-shelf cameras and ran them at relatively low frame rates, between 1 and 1.5 frames per second, relatively low resolutions, between 160 x 120 and 640 x 480 pixels, and normal 24 bit color. We changed the lens to a 120° field of view, wide-angle lens to increase coverage.

Tabulating video without abstraction is equivalent to representing each pixel as an independent variable across time. For typical video, this representation is a time series with several hundred thousand

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 13:07 from IEEE Xplore. Restrictions apply.

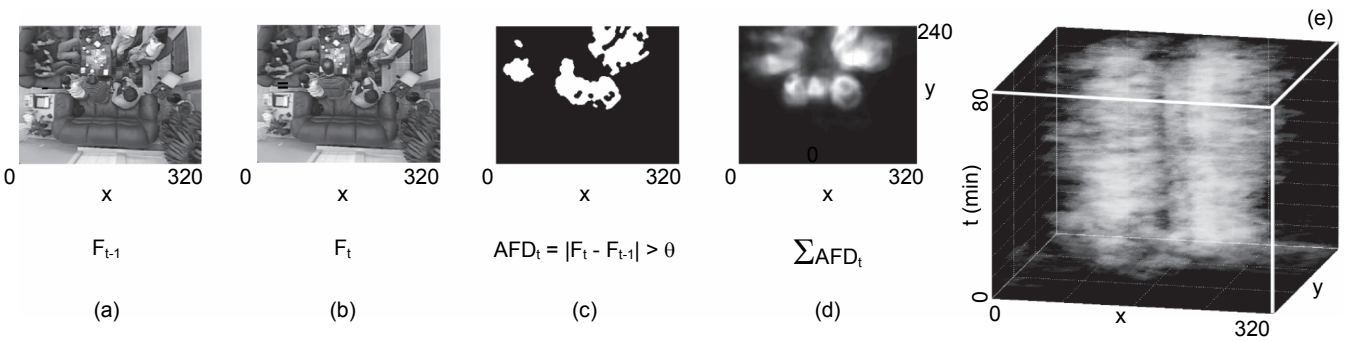

Fig. 3. Computing motion by adjacent frame difference (AFD). (a) Previous frame, (b) present frame, (c) adjacent frame difference (AFD), (d) sum of AFD over time, (e) activity cube shows motion over space (living room) and time (80 minutes) in 3D.

variables that is prohibitively obscure to analyze in practice. In practice, for overhead video, each pixel is not an independent variable. It shares high correlation with its spatial and temporal neighbors. Furthermore, the vast majority of pixels are irrelevant most of the time because nothing changes in their field of view. We take advantage of these inherent properties of overhead video to automatically compute simple and robust low level abstractions.

Of the abundant computer vision techniques, we purposefully chose to restrict our abstraction to motion. Motion is considered one of the most robust and lowest level abstractions from video [24]. Overhead video readily affords a number of important technical simplifications. First, the camera is fixed, both in its internal and external parameters (focal length, position, and orientation). Second, the optical axis is vertical. These two simplifications mean that we can, in practicality, assume there is a one-to-one correspondence between image and architectural space and that there is a single plane of interest, the ground. Ignoring the slight error introduced by parallax, mapping pixels to small areas in physical space is a simple, realistic, and robust abstraction. Third, in natural settings, changes in architectural space (image background) are rare events. Fourth, dramatic illumination changes occur very sporadically. Fifth, the likelihood of people appearing identical to the background is extremely low. At least some part of their body will be of a different color, shade or texture than the background. And sixth, the likelihood of people holding perfectly still drops to zero very quickly. Under these practical conditions, we compute motion from video by simple adjacent frame difference (AFD) [25] and we associate this motion with the physical space it occupies.

We subtract gray-scaled adjacent frames in time (Figures 3a-b) and threshold the difference (Figures 3c). The result is a binary motion image, where white pixels represent motion. We clean up the binary image with the morphological operators open and close. The threshold and the morphological operators serve as signal-to-noise ratio control parameters.

The binary motion image is significantly more compact than the original frame, yet it contains most of its semantic relevancy. It shows when, where, and how much motion occurred. As a concrete example, consider a 640 x 480-pixel, 24-bit frame. It contains 7,372,800 bits. A binary motion image of the same resolution contains 307,200 bits. Typically, motion is sparse. Assuming 5% percent of the pixels are active, a typical motion image can be encoded in a sparse matrix with roughly 15,000 bits. This is an abstraction that hides roughly 99.8% of mostly irrelevant data.

Since image space has a one-to-one correspondence with physical space, we can easily aggregate the data over space and time (figs. 1f-g, 3d, and 4b-h), and we can stack the motion frames so that time is represented in the third axis of a motion cube (Figures 3e and 4a). We call this cube the *activity cube*. It encodes the motion of people across image space, physical space, and architectural space across time. The activity cube and the aggregates we compute from it serve as the basis for our visualization (Figures 1g).

In Figure 4 we show our first-stage model of visualization and interaction with the activity cube. As with other 3D visualizations, the cube presents a number of challenges. Because of perspective and occlusion, to get a clear picture of the structures, we need to be able to rotate, translate and zoom the view in three dimensions.

We use the cube as a high level overview to the data and provide a number of marginal aggregations that serve as 2D and 1D "x-rays" of the cube (Figures 4b-h). These aggregates are higher abstractions of the data. Next, we augment these aggregate views with dynamic querying capabilities through double sided sliders. Finding target events in the cube is equivalent to defining the relevant spatial and temporal boundaries of a sub-space or manifold inside the cube (Figure 4i). At this stage, the only possible shape of the sub-space is an orthogonal parallelepiped. In reality, finding target events may require following translating motion across space. These types of events would be snake-like 3D manifolds inside the cube. Simple orthogonal query sliders are unable to capture such structures. To coarsely achieve this, a first approach is to augment the conjunctional queries with disjunction capabilities.

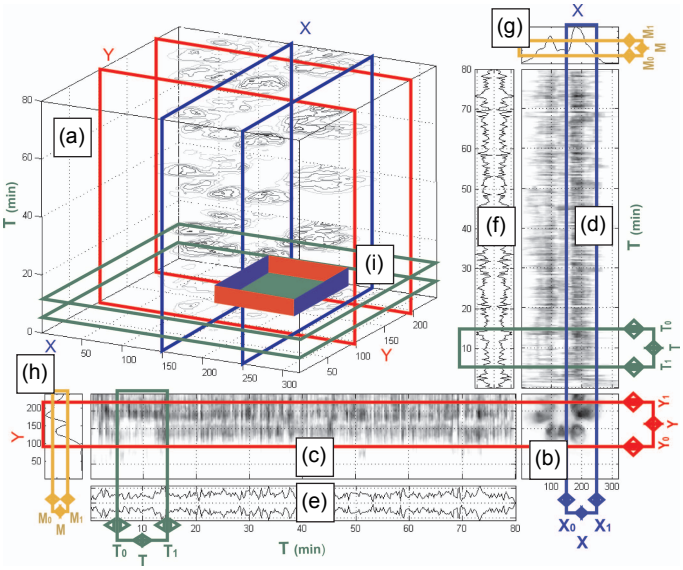

Fig. 4. Model of visualization and navigation for the activity cube. (a) Activity cube showing 5 aggregate 2D isocontour slices of motion across 80 minutes, (b) aggregation of motion across entire 80 minutes, (c) aggregation of motion across X (Y vs. T), (d) aggregation of motion across Y (X vs. T), (e-f) aggregation of motion across X and Y, (g) aggregation of motion across Y and T, (h) aggregation of motion across X and T, (i) sub-space result of the query $(X_0<X<X_1)\&(Y_0<Y<Y_1)\&(T_0<T<T_1)$. The dynamic query is performed through double sided sliders on X (blue), Y (red), and T (green). The fourth querying dimension is aggregate motion M (yellow).

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 13:07 from IEEE Xplore. Restrictions apply.

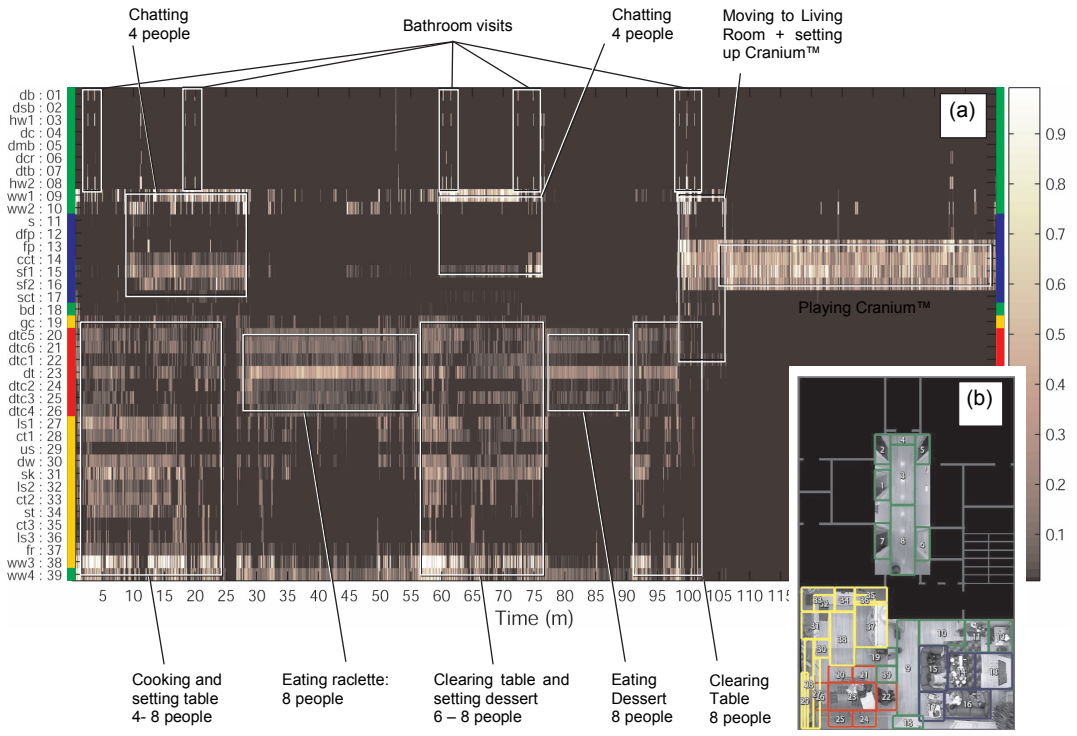

Fig 5. Semantic aggregation of motion: the Activity Table (a) and the Aware Home floor plan (b) with overhead images and Semantic Activity Zones (SAZ) on top. The activity table's rows visualize the level of motion over the places of the home across time. We map aggregate motion to brightness. Because spatial semantics are maintained, large spatial movements are clear across the table. The data presented is a dinner party with 8 people. We annotated some episodes during the 150-minute dinner party.

So far, we have presented purely spatial and temporal abstractions. These abstractions segment relevant semantics, but are not intrinsically semantic. The final level of abstraction we present in this paper is aggregation over places of interest. We define places (or regions) of interest manually. They could be defined dynamically and automatically, but we wanted to keep control of this process with the human at this first stage. We segment image/physical space into meaningful regions. We start with the observation that place is socially meaningful space. Our first method is to divide the image space into architectural elements of the space, such as hallways, doorframes, chimneys, kitchen counters and appliances. This is equivalent to segmenting the activity cube into pre-defined orthogonal parallelepipeds spanning the height of the cube. Next, we divide the space based on large furniture such as the couch, the coffee table, the dining table. We call these divisions semantic activity zones (SAZ) [26]. In all our observations these definitions remained stable throughout the deployments, even up to 6 months. If the furniture layout changes, though, there are simple computer vision algorithms to detect and track those changes. The furniture has fixed appearance since its distance to the camera remains relatively constant and there are no out-of-plane rotations. We did not address this automatic tracking since our deployments did not require it.

In Figure 5 we present the *activity table*. This version of the activity table maps the aggregate of motion over places of interest across time onto the intensity level of its rows across its columns, respectively. More generally, the activity table is a tabular representation of semantically aggregated motion across time. Figure 5b shows the floor plan of the Aware Home, Georgia Tech's living laboratory [27]. Figure 5b also shows the manual segmentation of the floor plan into SAZs. In this space we defined 39 zones. To highlight a couple of interesting examples, zone 15 is the living room sofa in front of the television that is mounted above the fireplace (zone 13). Zone 23 is the dining room table. The activity table in Figure 5a shows the activity of the 39 SAZs labeled on the left. The image streams come from 10 cameras, 4 in the living room, 2 in the dining room, 2 in the kitchen, and 2 in the hallway. We color coded the zones based on the regions they belong to: kitchen is yellow, dining room, red, living room, blue, and transit green. We added the color coding on the left and right edges of the activity table.

Note that the adjacency relationship between zones in the floor plan is two-dimensional. By aligning the zones along a single column, some adjacency relationships are lost. For example, zone 9 is adjacent to 8, 10, 15, 18, 19, 22, and 39. In the table, it is adjacent to 8 and 10 only. Thus, in order to visually track changes in location it is necessary to skip rows. This can be mitigated by row re-ordering or hiding. The problem with reordering and hiding is that part of the process of learning to read activity in this table relies heavily on row stability.

The data shown on this instance of the activity table is a dinner party of eight adults. They prepared dinner, ate, cleaned up, and played a game board in the living room. The data that we have shown in Figures 1 through 4 come from the lower right camera in the living room and from the period where the 8 adults played cranium.

The activity table is highly abstracted. It allows us to visualize 3 hours of data coming from 10 cameras at 1.5 fps and 320 x 240 resolution in a single 2D view. Without abstraction and excluding color, there are 768000 variables. With this abstraction, there are 39. We have eliminated 99.995% of the complexity. Of course, this reduction comes with a price.

The activity table is an effective visualization for large motions across space. The transitions between kitchen, dining room, and space are very apparent. We label this type of motion *translation.* The activity table, on the other hand, is not as an effective visualization for motion that does not produce a change in location. We call motion that occurs over the same space *vibration*. It is hard to distinguish fine events inside the large episodes annotated in Figure 5a. For example, during the game of Cranium™, there is a finer granularity that is lost in this visualization. The game has turn taking, it has different modalities of play, and it has different outcomes at each turn. All of these behaviors are washed out at this level of abstraction.

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 13:07 from IEEE Xplore. Restrictions apply.

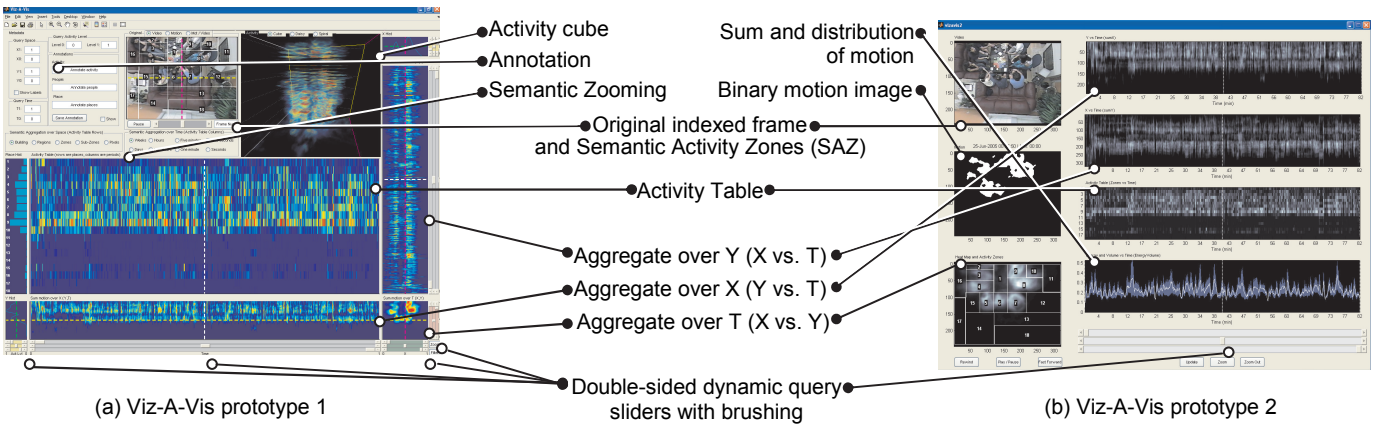

Fig. 6. Viz-A-Vis formative evaluation prototypes.

We tried several techniques to avoid losing sight of vibrations, including zooming and finer granularity for the parsing of space, a type of semantic zooming. These techniques help, but are not enough. We now present the process of reification, the practice of going from abstract to concrete representations.

3.2 Process of Interactive Reification

Up to this point, the only input from the analyst is the definition of semantic activity zones. We now describe in detail the types of exploratory interactions we designed for Viz-A-Vis, which serves as a reification toward the relevant raw data. At the abstract level users make hypothesis that they reify and test by looking at the original video.

Figure 7 shows the final interface for Viz-A-Vis. It is a geographical information system (GIS) where the geography is the floor plan of the environment, annotated with simple outlines of the furniture and spaces contained within it. The layers stacked on top of the floor plan are aggregate slices of motion across time.

The data in Figures 6 and 7 come from the bottom right camera in the living room during the episode of playing cranium at the end of the events in Figure 5.

This GIS-style visualization is the third prototype of a sequence we formatively evaluated through interviews with 8 information visualization researchers. We presented the three prototypes to each expert, explained the data, the analytical goals, the transformations and the views. The first prototype unfolded the orthographic aggregates horizontally and vertically (see Figure 6a) and downplayed the view of the cube in preference of the activity table. All but one of the reviewers found integrating the vertical and horizontal views of time awkward. The second prototype showed all aggregates across time horizontally, from left to right. The downside of this is that the X vs. T aggregate view is transposed and maps left to up and right to down (Figure 6b). Integrating the spatial information continued to be a challenge. We arrived at our GIS visualization for two main reasons: first, the visual integration of the aggregate views is simpler under 3D perspective; second, the floor plan provides valuable context for visually disambiguating the activity cube and its aggregates.

We will now review the design of the third prototype. First, we provide high level overviews in the activity table on the left and the activity cube. The activity table is not part of the 3D structure and sits in front of the cube. Rotations and translations do not affect the table. The user can brush space, place and time on both views, though, and zooming and filtering on either will affect the other and all the other views of the orthogonal aggregations. The activity table in Figure 7 is a transpose of the table in Figure 5. The SAZs are the columns of this table, and time goes from bottom to top, in the same direction of the cube. It seems more natural to show time starting at the ground and advancing up without boundaries.

Directly on top of the ground we show a heat map aggregating the entire time period being considered. Together with the outline of the floor plan and the furniture on it, this temporal aggregate serves as an effective summary of the activity during the time period at hand. Unfortunately, it hides the sequence of events. There are techniques that show aggregate and sequence of motion, for example, temporal templates [28]. This technique fades the motion as time goes by. Unfortunately, it does not scale well for long and complex sequences where multiple actors occupy the space under observation.

Separated by a prudent gap to avoid occlusions, the activity cube lies directly above the temporal aggregate and the architectural space it tracks. Here, we are showing the same data as in Figure 3e and 4a. Since the motion captured in this video sequence is vibration, the activity cube naturally forms cylindrical columns in the places where people sat.

We aggregate the data into roughly one-minute slices. The temporal window of aggregation is an important parameter of the visualization. Different temporal patterns will emerge at different aggregation granularities. Some patterns will emerge with a two-second aggregation window, like loading the dishwasher, while other patterns will emerge with a one-day granularity, like weekdays versus weekends. Furthermore, the number of temporal slices is constrained by the space and resolution of the display screen. For Viz-A-Vis we compute by default a discrete optimal aggregation window as a function of the length of the sequence and the size of the screen. We also allow the user to manually define the aggregation window if needed. We double map each heat map layer in the cube to color and opacity. Thus, areas with lower aggregate values will be simultaneously darker and more translucent. We experimented with several views, including, voxel representations, isocontours, and isosurfaces. Translucent aggregate slices maintained the visual structure of the data better than the other options.

On the "walls" of the GIS we show the aggregate of motion across X and Y. They serve as x-rays of the activity cube. The offer navigation and contextual affordances through brushing and dynamic querying over time.

We've extracted the original frame and the binary motion image at the temporal point of brushing. This rapid indexing provides detail and focus and maintains the temporal and spatial context. It lets the user interpret the video data from the source. The images are laid out horizontally, as if cards drawn from a deck. The user has the option of hiding this detail. The analyst can brush the cube and pull out the original data by scrubbing with the mouse over the temporal brush. We provide typical video playback capabilities as well.

On the left hand side of figure 7 are three 2D graphs: the activity table, the aggregate and dispersion of motion, and the heat map with the semantic activity zones overlaid. The heat map of activity aids the user define the regions of interest in the X-Y plane. It provides a high level view of real usage patterns over the space of interest. Together with the floor plan, they help discover the real and dynamic

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 13:07 from IEEE Xplore. Restrictions apply.

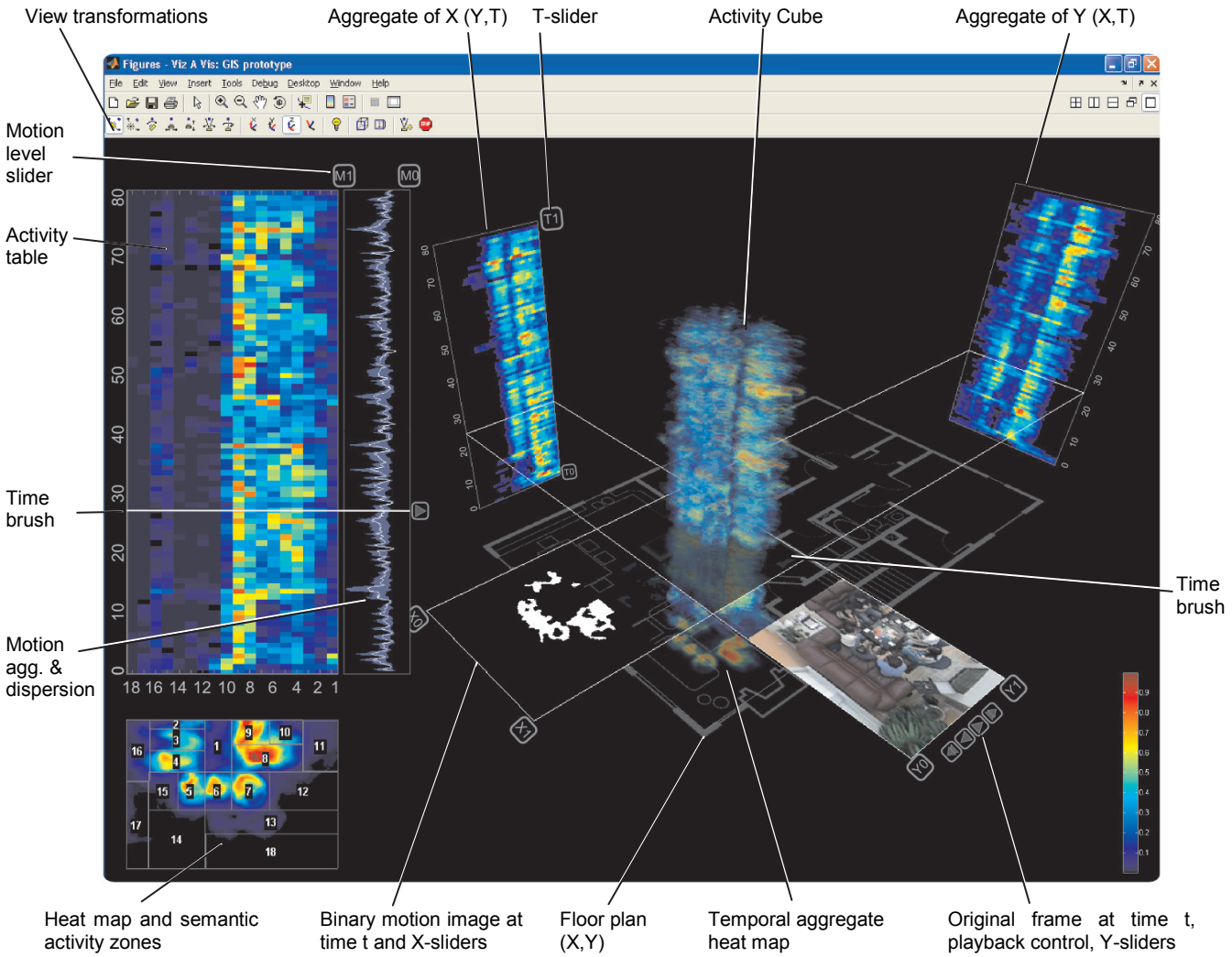

Fig. 7. Viz-A-Vis interface. *Overview* : Activity Table, Activity Cube. *Zoom*: double-sided sliders for dynamic query on time and space. *Filter*: motion level double-sided sliders, cube translucency, and opaque time brush surface on cube. *Detail, index* and *focus*: binary motion image and original frame at time t with playback controls. *Context*: floor plan, activity cube, temporal and spatial aggregates. *Temporal aggregation:* heat map. *Spatial aggregation*: X vs. T and Y vs. T. *Semantic aggregation*: semantic activity zones definition and activity table. *Semantic Zooming*: activity table. *Brushing*: time brushing. *View transformations*: 3D-view rotate and translate, camera roll, pitch, yaw, position, and field of view, and variable illumination from multiple lights.

social semantics of architectural space.

We conclude this section with a description of the line-and-area plot of the aggregate and dispersion of motion on figure 7. The white line in the plot encodes the aggregate of motion over the entire space of observation. It is a very high level summary of the amount of activity in the scene. The plotted blue area in the same axis encodes the *dispersion* of motion over the semantic activity table. It measures how compact or disperse the motion is. It helps differentiate similar motion aggregates resulting from different behaviors. For example, a single person moving rapidly may generate the same motion aggregate as numerous people moving slowly. The dispersion of multiple people will be higher. We approximately compute dispersion by thresholding the activity table and summing the pair-wise distances between non-zeros elements. This definition and approximation to dispersion is one example of higher level semantics from computer vision and pattern recognition. Together with the motion aggregate, these abstractions have proved instrumental in the analysis of this time series.

4 Preliminary Case Study of Viz-A-Vis: VRP Occupancy

We present a preliminary case study of applying Viz-A-Vis to understanding behavior. The study explores the effect of three different projection technologies on groups of people collaboratively interacting with a projection surface. We report our application of Viz-A-Vis to the problem of understanding the effect of three different Virtual Rear Projection (VRP) technologies [29] on a collaborative group of users working with an interactive projection surface. The goal of VRP is to simulate the experience of true rear projection without sacrificing the physical space necessary for it. A VRP system aims to eliminate shadows on the projection surface and prevent light from falling on objects (such as users) other than the projection surface.

Figure 8 (top row) illustrates the three experimental conditions: Single Projector (SP), Passive Multiple Projector (PMP), and Active Multiple Projector (AMP). SP and PMP simply mitigate shadows on the surface by off-center projection and redundancy. Only AMP corrected for shadows on the board *and* for light falling on other objects.

In the study, five groups of three to five people were asked to work on a collaborative task at a large interactive display for fifteen minutes, split into three five-minute sessions, one for each projection technology. We recorded overhead video for each condition, recorded camcorder video with audio for manual analysis, and

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 13:07 from IEEE Xplore. Restrictions apply.

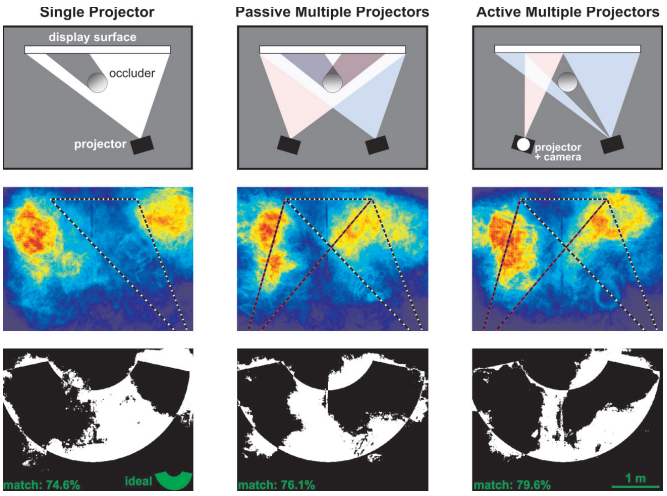

Fig. 8. Viz-A-Vis summary visualizations for VRP. The three columns correspond to the three testing condition of Virtual Rear Projection. The first explains each technology. Row two visualizes aggregate motion heat maps. The third row visualizes template matching to ideal model. The percentages correspond to the match.

collected self report data from questionnaires and interviews.

We explored the data through the different spatial, temporal, and semantic aggregations of Viz-A-Vis. The aggregate that revealed the most interesting and succinct patterns was the temporal aggregate heat map over the space in front of the projection surface. We show this heat map for each condition across the second row of Figure 8. The heat maps revealed trends that were not visible when watching the groups operating live in real-time, through a camcorder recording, or even through manual analysis of the raw overhead video.

In the SP condition (left column), users are clearly split by the projected light (entering diagonally from the bottom right towards the SmartBoard located at the top center) which results in the large (blue) area showing minimal activity near the middle of the room. The people to the right of the projector beam are standing forward, towards the wall and away from the projected light. The PMP and AMP conditions also show a bi-modal distribution, but those groups are much closer together, and when compared to the SP condition, the right group is not pushed as far forward. Part of the functionality of Viz-A-Vis is to be able to take individual views and extract them from the GIS. Being able to see the aggregate motion side by side, organized by condition, allowed us to notice that the AMP condition appeared to be even less split than the PMP condition.

From this visualization we derived the concept of an “ideal" model of space usage for collaboration and used this model to quantify the space usage for numerical comparison. As we stated at the start of the paper, our third goal for the Viz-A-Vis approach is to find new features and patterns that can improve the computer vision. The ideal model we describe here is an instance of a visual pattern we discovered which can be used to advance the computational perception.

We noted that users in all three conditions where approximating a semicircular arc before the SmartBoard. We developed an "ideal" space usage model, the semicircular arc shown superimposed on the bottom row of Figure 8, because 1) the hole in the center allows all users equal view and physical access to the board, and 2) the circular shape also allows equal social access to other participants. This arc is an abstraction step chosen by the analyst, a deliberate introduction of bias to gain rapid abstraction. We used a template match by sum of square differences (SSD) to compare the actual study data to the semicircular arc model.

SSD is a metric of the difference between the average activity in each condition and the ideal model. This calculation is shown graphically in the bottom row of Figure 8. As the conditions’ match-to-ideal progress from SP (74.6%) to PMP (76.1%) and AMP (79.6%), the occupancy approaches the abstract ideal. This monotonically increasing value surprised us, since the totality of user self report preference data ranked PMP well above the other conditions. The ability to aggregate user motion over time allowed us to understand how the projection conditions affected user's space usage, develop a mental model of an "ideal" space usage pattern based upon actual data, and discover that user behavior in the AMP condition matched this model closer than in the PMP condition. This analysis motivates further study of the behavioral differences between the PMP and AMP conditions. In this application domain Viz-A-Vis enhanced the analysis of previously clouded phenomena of human behavior.

5 Conclusions and Future Work

The central theme of this paper has been to introduce computer vision and pattern recognition as an automatic augmentation to the low level transforms that convert raw data to data tables. The *activity table* is a dramatic example of this approach. In it, we segmented and aggregated to less than 0.005% of the raw variables for the high level abstract visualization. We demonstrated the power of automatic abstraction through computer vision and we gave a step toward explorative reification through information visualization. We recognize this is just a first step. Computer vision, statistical machine learning, and pattern recognition have a profound and diverse set of tools that can be applied to this domain. Our key observation is that while high level reasoning has remained elusive for computers, low level data transformation can be achieved robustly with current methods. Moreover, low level data integration is not an efficient task for humans. The human analytical task supported by information visualization is augmented by robust low level computational perception. This is one of the most sought after paradigms of human computer interaction. Each part of the human-computer system performs the task that responds to its strengths.

We conclude with a discussion of future work, which we consider to be extensive. Viz-A-Vis was born out of our work building ubiquitous computing systems. In our deployments, we are constantly sensing the context in which the computing systems perform. In the process of building and testing our perceptive and interpretive infrastructures, we are routinely building visualizations of the low level sensor data. The activity table is an instance of a visualization we built for designing context aware systems. The reiterative need for contextual visualizations of sensor data was our original motivation for building Viz-A-Vis. In this paper we have concentrated on video data due to its complexity, acuity, extensive volume, and the large semantic gap between the raw data and high level understanding of events over time. Basically, it is very hard for humans to objectively perceive in video extended spatiotemporal patterns, such as occupancy. Even if an analyst watches the original video it’s unlikely that the objective patterns of occupancy will become evident. The analyst needs exterior tools, such as parsing the image and counting locations on paper and pencil.

Although we have only visualized video data, the GIS infrastructure of Viz-A-Vis allows multimodal sensing and visualizing. We are working on integrating different sensing modalities to the visualization.

In bridging the semantic gap from the bottom-up, we have just given a first exploratory step. The natural next computer vision steps up the semantic ladder are background subtraction and maintenance, blob tracking, object and human detection, tracking, and recognition, region-of-interest discovery, and activity discovery and recognition. From machine learning and pattern recognition the next steps are k-mean or radial basis function automatic clustering of space, principal and independent component analysis of the raw data, interactive feature generation, adaptive boosting, Hidden Markov Models and dynamic time warping. While we increase the semantic abstraction at each step, we introduce new complexity and brittleness to the

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 13:07 from IEEE Xplore. Restrictions apply.

human-computer system. Given the current state of the art in computational perception, the analyst will eventually lose faith in the abstractions. And even if the case were that the abstraction are indeed perfect, because of the nature of the task and data, the analyst should always have direct access to the original data.

As the higher level semantics from computational perception are abstracted from the raw data, new visual structures need to take advantage of the affordances presented. For example, what does it mean to have blob tracking for the activity cube and table, which are essentially space tracking representations? Finally, there are opportunities for creating infrastructure to allow the user to defined patterns and the machine to search for them using, for example, dynamic time warping and template matching techniques. We need to continue to explore these venues through an iterative design and evaluation process. As Viz-A-Vis and its methodologies mature, we will have an opportunity to comparatively evaluate its measurable benefits in terms of precision, recall, and time to task completion.

REFERENCES

[1] S. Card, J. Mackinlay, and B. Shneiderman, *Readings in information visualization: using vision to think*. San Francisco, Calif.: Morgan Kaufmann Publishers, 1999.

[2] M. Pantic, A. Pentland, A. Nijholt, and T. Huang, "Human Computing and Machine Understanding of Human Behavior: A Survey," in *Artifical Intelligence for Human Computing*, vol. 4451/2007, *Lecture Notes in Computer Science*: Springer Berlin / Heidelberg, 2007, pp. 47-71.

[3] C. Marshall and G. B. Rossman, *Designing Qualitative Research*, Fourth ed. Thousand Oaks: Sage Publications, 2006.

[4] K. Miller, *Principles of Everyday Behavior Analysis*, Fourth ed: Wadsworth Publishing, 2005.

[5] S. R. Barley, "Images of Imaging: Notes on Doing Longitudinal Field Work," *Organization Science. Special Issue: Longitudinal Field Research Methods for Studying Processes of Organizational Change*, vol. 1, pp. 220-247, 1990.

[6] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 3, pp. 3, 2007.

[7] P. Stéphane, G. Pierre, and P. Justin, "Prototyping of interactive satellite image analysis tools using a real-time data-flow computer," in *Image Analysis and Processing*, vol. 974/1995, *Lecture Notes in Computer Science*, 2006, pp. 683-688.

[8] A. Wilson, "Robust Vision-Based Detection of Pinching for One and Two-Handed Gesture Input," presented at UIST, 2006.

[9] K. Truong, G. Abowd, and J. Brotherton, "Who, What, When, Where, How: Design Issues of Capture and Access Applications," in *Proceedings of the 3rd international conference on Ubiquitous Computing*. Atlanta, Georgia, USA: Springer-Verlag, 2001.

[10] J. Hare, P. Lewis, P. Enser, and C. Sandom, "Mind the Gap: Another look at the problem of the semantic gap in image retrieval," presented at Multimedia Content Analysis, Management and Retrieval 2006, San Jose, California, 2006.

[11] G. Hayes, "Documenting and understanding everyday activities through the selective archiving of live experiences," in *CHI '06 extended abstracts on Human factors in computing systems*. Montreal, Quebec, Canada: ACM, 2006.

[12] C. Neustaedter, S. Greenberg, and M. Boyle, "Blur filtration fails to preserve privacy for home-based video conferencing," *ACM Trans. Comput.-Hum. Interact.*, vol. 13, pp. 1-36, 2006.

[13] V. Bellotti and A. Sellen, "Design for Privacy in Ubiquitous Computing Environments," presented at Proceedings of the Third European Conference on Computer Supported Cooperative Work ({ECSCW}'93), 1993.

[14] G. Iachello and G. Abowd, "Privacy and proportionality: adapting legal evaluation techniques to inform design in ubiquitous computing," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. Portland, Oregon, USA: ACM, 2005.

[15] Y. Ivanov, C. Wren, A. Sorokin, and I. Kaur, " Visualizing the History of Living Spaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1153-1160, 2007.

[16] T. Kapler and W. Wright, " GeoTime Information Visualization," presented at Information Visualization, INFOVIS 2004, Austin, Texas, 2004.

[17] M. P. Kwan and J. Lee, "Geovisualization of human activity patterns using 3D GIS: a time-geographic approach," in *Spatially Integrated Social Science: Examples in Best Practice*, M. F. Goodchild and D. G. Janelle, Eds. New York: Oxford University Press, 2004, pp. 48–66.

[18] G. Daniel and M. Chen, "Video Visualization," in *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*: IEEE Computer Society, 2003.

[19] A. W. Klein, P.-P. J. Sloan, A. Finkelstein, and M. F. Cohen, "Stylized video cubes," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. San Antonio, Texas: ACM, 2002.

[20] S. Fels, E. Lee, and K. Mase, "Techniques for interactive video cubism (poster session)," in *Proceedings of the eighth ACM international conference on Multimedia*. Marina del Rey, California, United States: ACM, 2000.

[21] E. P. Bennett and L. McMillan, "Proscenium: a framework for spatio-temporal video editing," in *Proceedings of the eleventh ACM international conference on Multimedia*. Berkeley, CA, USA: ACM, 2003.

[22] M. Terry, G. J. Brostow, G. Ou, J. Tyman, and D. Gromala, "Making space for time in time-lapse photography," in *ACM SIGGRAPH 2004 Sketches*. Los Angeles, California: ACM, 2004.

[23] S. Kiranyaz, K. Caglar, E. Guldogan, O. Guldogan, and M. Gabbouj, "MUVIS: a content-based multimedia indexing and retrieval framework," presented at Seventh International Symposium on Signal Processing and its Applications, ISSPA 2003, Paris, France, 2003.

[24] A. Bobick, "Movement, Activity and Action: the Role of Knowledge in the Perception of Motion," *Royal Society Workshop on Knowledge-based Vision in Man and Machine*, vol. 352, pp. 1257-1265, 1997.

[25] M. Sonka, V. Hlavec, and R. Boyle, *Image Processing Analysis, and Machine Vision*: PWS Publishing, 1999.

[26] M. Romero, Z. Pousman, and M. Mateas, "Alien presence in the home: the design of Tableau Machine," *Personal and Ubiquitous Computing*, vol. 12, 2007.

[27] C. Kidd, R. Orr, A. G., A. C., I. Essa, B. MacIntyre, E. Mynatt, T. Starner, and W. Newstetter, "The Aware Home: A Living Laboratory for Ubiquitous Computing Research," presented at Proceedings of the Second International Workshop on Cooperative Buildings - CoBuild'99, 1999.

[28] A. Bobick and J. Davis, "Real-time recognition of activity using temporal templates," presented at 3rd IEEE Workshop on Applications of Computer Vision (WACV '96), 1996.

[29] J. Summet, M. Flagg, T. Cham, J. Rehg, and R. Sukthankar, "Shadow elimination and blinding light suppression for interactive projected displays," *IEEE Transactions on Visualization & Computer Graphics (TVCG)*, vol. 13, pp. 508-17, 2007.

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 13:07 from IEEE Xplore. Restrictions apply.