The ContextCam: Automated Point of Capture Video Annotation

Shwetak N. Patel and Gregory D. Abowd

College of Computing & GVU Center
Georgia Institute of Technology
801 Atlantic Drive, Atlanta, GA 30332-0280, USA
{shwetak, abowd}@cc.gatech.edu

Abstract: Rich, structured annotations of video recordings enable interesting uses, but existing techniques for manual, and even semi-automated, tagging can be too time-consuming. We present in this paper the ContextCam, a prototype of a consumer video camera that provides point of capture annotation of time, location, person presence and event information associated to recorded video. Both low- and high-level metadata are discovered via a variety of sensing and active tagging techniques, as well as through the application of machine learning techniques that use past annotations to suggest metadata for the current recordings. Furthermore, the ContextCam provides users with a minimally intrusive interface for correcting predicted high-level metadata during video recording.

1 Introduction

An ambition of ubiquitous computing is to create services that perform some of the important yet mundane activities that we would like to do, but do not have the patience or time to perform. One of those tasks is annotation, indicating the salient pieces of information that describe a situation for future reference. A richly annotated family history —pictures, movies, physical artifacts all accompanied by descriptions and narrative that describe their significance— could provide a variety of entertaining and valuable services, but we rarely have the time to do the annotation. There is an opportunity for technology to perform the annotation on behalf of the user.

What would it take to have the ability *today* to record live events and tag them at the point of capture with metadata that can be preserved and aid in review? We present a solution to this problem for live video, called the ContextCam. We developed the ContextCam to solve a problem of simplified annotation of home movies, motivated by a previous mainly manual annotation system we built call the Family Video Archive (FVA) [1], and shown in Figure 1. As we will show, annotation of home movies provides a very meaningful way to share large archives of family history, but only when metadata describing when, where, who and what activity is taking place is added. Although this data can be added manually, as our previous work demonstrated, the real challenge is to automate as much of this tedious task as possible.

N. Davies et al. (Eds.): UbiComp 2004, LNCS 3205, pp. 301-318, 2004.

In this paper, we describe how the management of home movies motivated the development of the ContextCam, and discuss related work on automated annotation of digital memories. We then describe the prototype ContextCam device, a digital video recording device with a collection of sensing techniques to annotate video with information concerning when and where the video is being recorded and, more importantly, who is in and around the field of view of the camera. This latter metadata is accomplished through an active tagging technique. We then show how the low level metadata for *who, where* and *when* can be used to infer higher level information with a Bayesian prediction technique that supports a manual technique for assigning higher-level semantic tags to a scene while recording. We also present the user interaction with the ContextCam that integrates the point of capture annotation with the regular use of the augmented video camera. Although the ContextCam was designed specifically to address an opportunity for home movies, its application extends that entertainment motivation.

Figure 1: Manual tagging interface for the Family Video Archive [1].

2 Motivation: A Semi-automated Family Video Annotation System

Historically, families have recorded their history using visual aids such as paintings, photographs or videos. With digital technologies for photography and videography, the storage and manipulation of family history is easier than it has ever been. In addition, families are even beginning to convert old 8mm and VHS films into a digital form. Although tools exist to organize and share digital photographs, the only tools

available for digital video are ones for nonlinear editing of video sequences. Despite the increasing relative ease of recording and editing family movies, archiving and retrieving them is not quite as effortless. For instance, consider the overwhelming task of finding and creating a video compilation of a family's Christmas scenes from the past 20 years. This would be a wonderful gift to share with loved ones, but it is often a very time-consuming task to assemble the individually recorded video snippets into a complete presentation. We earlier designed a system, the Family Video Archive, that supports the manual annotation and browsing of a large collection of home movies [1]. Motivated by PhotoMesa's [3] zooming interface for browsing digital photographs, the system provides a flexible semantic zooming interface for browsing and filtering the collection of user-defined scenes. Related scenes can be grouped into albums and exported out of the system, either as input to a nonlinear video-editing suite or as an independent DVD with menus and navigation. These browsing and authoring features will not be described in any further detail in this paper. Instead, we focus on the annotation features of our system, the foundation for any of these browsing and authoring capabilities.

2.1 Annotating Home Video

In our earlier home video annotation system (see Figure 1), users import video files into the system by dragging them into the video files browser in the lower left corner of the interface. Users view video in the video playback panel using a VCR style control interface. A timeline indicates the user-defined scene boundaries within any given video file (there is no assumption that video files contain single scene information). Using a simple frame-by-frame color histogram comparison, the system suggests scene boundaries that the user can accept, modify or remove.

The interface provides a variety of ways to attach metadata to a video scene, and the accumulated metadata is shown in the top middle section of the interface. Date, freeform text, and user-defined metadata tags are possible annotations within this system. Although the metadata categories are completely user-defined and form a potentially complex hierarchical network, we expect that in any home video archive three major categories of interest will emerge: *Who* is in the scene, *Where* that scene is and what *Event* is depicted in the scene. There are a few ways to associate a metadata tag to a video scene, the simplest being to drag the tag from the hierarchy on the right of the interface to the video window.

The freeform text window at the top of the interface is a quick way to associate arbitrary text to video scenes, making it a convenient interface when people are talking aloud about the video during a family gathering. As we describe next, this freeform text can be used to help accelerate assignment of metadata tags to a scene.

2.2 The Need to Improve Annotation Efficiency

Our experience with managing home video motivates the automatic annotation of *who, what, when, where* at the point of capture. Rich and accurate metadata is the key to managing a large video archive. As video content and potential tags grow, it becomes increasingly time consuming to annotate video scenes. Recognizing the

trade-off between time and richness of annotation, we designed features available in our original system to speed up annotation without sacrificing correctness. First, the semi-automatic scene detection mentioned earlier can save considerable time by inferring scene changes, but can vary with the quality of the video recording. Second, the application matches the free-form text description against the names of metadata tags, placing suggested tags in a separate part of the annotation interface. The tag suggestion window contains far fewer tags than the tagging hierarchy window making it easier to find appropriate tags. A prioritization function places more likely tag candidates near the top of the list to make things even easier. Third, the same tag can be assigned to multiple scenes. Often, when viewing some scenes, a user suddenly becomes aware of a new piece of information (*e.g.* a new person is identified or the location is clarified) that applies to several of the preceding scenes. Rather than requiring users to load each of those scenes and add the new annotation, users can apply this new tag to all of them simultaneously. Fourth, multiple tags can be assigned to a scene simultaneously by dragging a branch from the tagging hierarchy. For example, dragging the "My Family" category over a scene assigns every person under that category to that scene.

Despite these accelerators for annotation, users still require a large amount of time to do the annotation properly. We observed an expert (both with the video content and the annotation interface) using the annotation interface for one hour. In that time, he was able to annotate 30 minutes of video, using 255 metadata tags (54 unique ones), free form text, and date stamps. We identify the manual annotations in increasing order of occurrence per video scene:

- **Time/Date**: Time information is usually just specified once in a scene.
- **Event**: Scenes typically depict only a small number of events (usually just one) but there may be sub events within a video scene that a user wants to tag.
- **Location**: Location is typically assigned once in a scene but a variety of higher-level location tags may be assigned if the location is hard to determine or unfamiliar.
- **People**: There may be many people in a scene and the user may need to view all of the video to account for everyone.

Much of the tagging overhead is due to the user actually viewing and reviewing various segments of the video to acquire enough context information to make tagging decisions. From our case study, two-thirds (40 minutes) of the whole annotation time consisted of video playback and navigation. The overhead of physically searching and selecting tags accounts for the rest of the annotation time.

Annotation at the point of capture can potentially eliminate time, location and people tagging. The only annotation that remains is event information, which we will show can be inferred based on current time, location, people and past annotations. The problem of losing or forgetting important information because of retrospective manual tagging further motivates the point of capture approach. Manual tagging heavily relies on recalling and recognizing past events that may have faded from memory.

3 Related Work

The proliferation of digital artifacts, such as photographs and videos, has motivated many researchers and companies to address the general problem of collecting, organizing and browsing memories. Much work has been done for digital photographs, including automated methods for organizing based on feature analysis of the images [11, 19, 27], visualization strategies [2, 3, 13, 14, 21, 24, 28] and annotation techniques similar to what we described in the previous section [7, 16]. Other useful research (*e.g.* CMU's Informedia Project [30]) and commercial (*e.g.*, the VideoLogger from Virage [29]) tools show the promise of tools to browse and search digital video libraries.

However, there is still little work in automatically generating annotations for videos and some of the techniques used in automated photograph annotation do not easily extend to video. We identify four different annotation approaches.

Low-level semantic detection: Automated techniques exist for doing keyframe extraction and scene boundary detection [12], and many of these assume high-quality, structured video and/or textual metadata. These techniques provide little to no higher-level semantic meaning. Although possible with photographs, analyzing content of video is a much more computationally expensive task. Video annotation environments have been developed for detailed analysis of audiovisual data as well [8]. Researchers in linguistics and film studies perform fine-grained analysis of audiovisual data and tools like Anvil and IBM's VideoAnnEx support this kind of analysis [15, 30].

Manual annotation: Systems that provide manual annotation of high-level semantics for home videos are beginning to emerge [1, 4, 7]. These systems typically feature an interface where users can manually enter metadata as they review the video. In addition, television broadcasts are typically manually scripted and often synchronized with closed captioning.

Context acquisition at time of capture: Researchers are also now looking at point of capture tagging of photographs by sensing context information from standalone digital cameras and integrated camera phones [9, 27]. This usually consists of time and location stamping. In addition to spatio-temporal metadata, we are looking to add information about the people in the view of the camera (and nearby but not in the field of view) and to infer event information.

Predictive annotation: The Garage Cinema research group at UC Berkeley has recently produced a prototype point of capture system for camera phones [27]. Time and location information (at a very coarse level) are captured on camera phones and submitted to a network-based prediction algorithm that provides a web interface to select higher level semantic information that is used to feed a prediction system. Our motivation aligns with the work of theirs, which advocates very strongly the need to consider annotation at the point of capture. In addition to working with video as opposed to digital still imagery, we are also distinguished from this previous work

because we do not assume a network-based interaction for higher-level semantic annotation in our approach to the ContextCam.

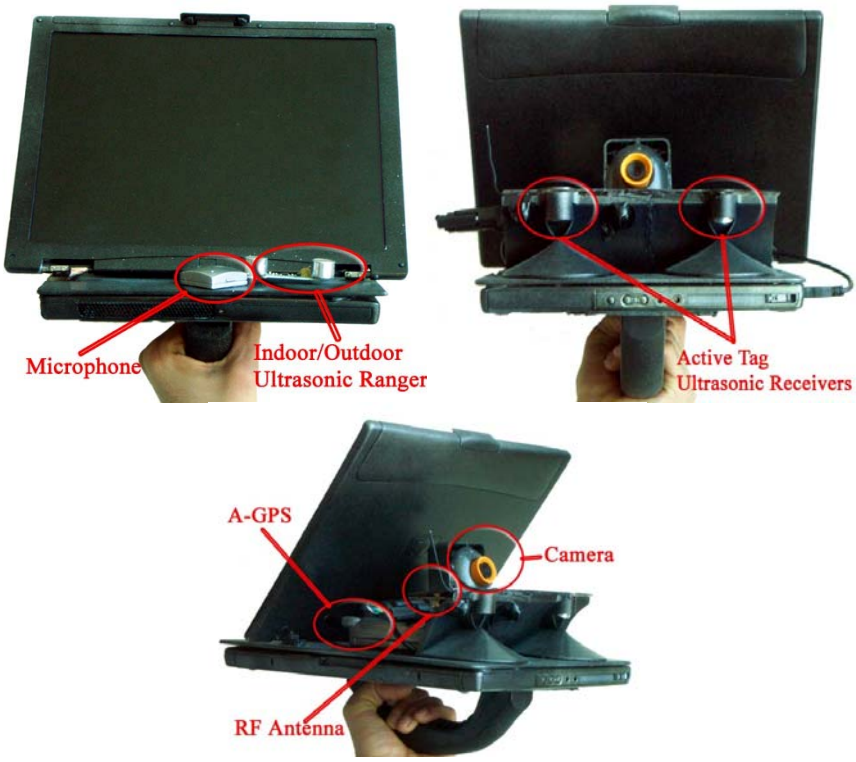

Figure 2: The ContextCam and its sensing features.

4 Low-Level Point of Capture Information for Annotation

Figure 2 shows the ContextCam prototype and some of its context sensing features. This original prototype is a modified Pentium-III laptop with camera and sensors mounted behind a viewing display. The software running on the ContextCam uses the Java Media Framework (JMF) for video capture and access. This prototype was designed as a proof of concept of the underlying technologies. In this section, we describe all of the sensing capabilities of the ContextCam that relate to simple, unambiguous metadata that can be associated to recorded video. In the next section, we demonstrate how this low-level context information, combined with past annotations, helps infer high-level semantic metadata.

4.1 When a Video Scene Is Recorded

The date and time of recording are fundamental and straightforward metadata to associate at the point of capture. Though many camcorders actually record date and time, most capture software does not access or preserve that information. The ContextCam acquires date and time information from an internal clock as well as from GPS satellites. As we will discuss later, this metadata is then encoded in the video itself for later extraction in our annotation/browsing environment.

4.2 Where a Video Is Recorded

To annotate low-level location information, the ContextCam uses a multi-channel assisted-GPS (A-GPS) receiver from a Motorola iDEN i730 handset. The assisted GPS systems provide slightly better accuracy and reliability than standard GPS receivers. We were able to get fixes where a traditional GPS receiver could not, such as some indoor locations and near windows. The GPS provides raw latitude/longitude coordinates and is stored as metadata in the video stream, which is also converted and stored as higher-level semantic tags. However, there are many cases where the ContextCam cannot get a clear fix. We address this limitation by using cached location fixes from the cellular network. Although not as accurate, this still provides at least city level location information.

GPS and cached fixes do not provide information to distinguish between indoor or outdoor shots. Researchers have classified indoor and outdoor locations in photographs using color and texture analysis [19]. Because the ContextCam is present in the environment at the point of capture, it can sense whether it is indoors or outdoors directly. We use the reflection of an ultrasound chirp sent vertically up from the camera to determine if it is indoors. The ultrasound ranger is tuned to about 50 feet, so the lack of a reflection indicates outdoors and any detection indicates an indoor location. Although this solution does not work everywhere, it works for most situations. We considered other approaches such as 60 Hz artificial light detection, but found this ultrasonic solution was better.

4.3 Who Is in or Around the Field of View

One of the most time consuming tasks of manual annotations is tagging all of the people appearing in a scene. From our observation, this accounts for 75% of the annotation process. Retrospective tagging of people in a scene is very slow and does not guarantee that everyone will be recognized. Though the FVA annotation interface provides the ability to indicate exactly when within a scene a particular person was present, it is far too tedious to add this information, so the default setting is individual being “in” the video for the duration of a scene. Because knowing who is in a scene is important, we designed the ContextCam to provide a solution that can quickly and accurately determine this information. As an added bonus, our solution allows us to annotate who is nearby a scene but not in the field of view, something that is difficult to know when annotating retrospectively.

For accuracy and reliability we explore an active tagging approach for detecting people in or around the field of view of the camera. However, we considered passive techniques first. Computer vision would be computationally expensive and unreliable, being further complicated by relatively low quality images and occlusions that are part of home videos. Furthermore, a vision solution would only ever be able to determine people within the field of view of the camera, and we see an opportunity to sense people outside the field of view as well. Sound detection techniques would be even less appropriate for constant tracking of people. Passive tagging techniques, such as some forms of RF ID, would not be practical because of the range limitation for reading tags and the lack of precision for distinguishing tags within and outside the field of view.

We envision people who are part of a captured event wearing or carrying a small active token. The active tag is used to gather information about their position in relation to the camera (in its view or somewhere near it). These tags are low power and small enough to integrate into something that an individual is likely to wear, such as jewelry, or carry, such as a cellphone. Our current prototype tags are bulky and inappropriate for deployment (see Figure 3), but there is no inherent reason why they could not be designed to a more appealing form factor.

Figure 3: An active tag used to determine individuals in the view of the camera. A production tag could be much smaller and aesthetically pleasing.

We explored a variety of active tagging techniques such as Infrared (IR) and radio frequencies (RF). IR is typically limited to short range and is difficult to triangulate at longer ranges. IR is also susceptible to occlusions and is greatly affected by ambient light. RF, on the other hand, is not as completely affected by occlusions and generally provides long-range communication, but RF triangulation is prone to reflections and other conditions which limit its resolution and accuracy. Despite these drawbacks, RF is a good solution to detect people who are near or around the camera.

We chose an ultrasonic time of flight system for our location information. Ultrasound does not provide long-range communication like RF, but it provides very accurate location and position information. Although it also can suffer somewhat

from occlusion and multi-path complications, for our needs, we found that it is the best compromise between range and accuracy.

Partly inspired by MIT's Cricket [22, 23] and other location systems in the literature [10, 31], we use ultrasound to triangulate position of active tags around the camera. Each tag periodically chirps an ultrasound sequence and two ultrasound receivers mounted on the camera detect these chirps and calculate the relative distance between the tag and the receivers. The farther apart the receivers on the camera are, the finer the location resolution. For our prototype, we place the ultrasound receivers 6 inches apart, which yields position accuracy within 1 foot.

The ultrasound emitters on the active tags operate at 40 kHz, though less audible higher frequency emitters can be used. They are tuned to operate at ranges up to 50 feet. A higher-powered emitter could also be used for longer-range applications. A 300 MHz TE 99 Ming RF module accompanies the ultrasonic emitter on the tag. A single active tag emit cycle consists of a RF ping, followed by 5 ultrasound chirps. The RF ping, which consists of a 32 bit unique identifier, synchronizes the clock to the time the flight of the ultrasound chirp. Five ultrasound chirps spaced 10 ms apart follow the initial ping. The active tag continuously emits the RF/US sequence and draws at most 50 mA, which lasts about a day on a pair of 1.5-volt lithium ion cell batteries.

We place two ultrasound receivers on a horizontal plane in front of the camera. The cone shaped bases help direct the ultrasound chirp towards the receiver, increasing the likelihood of detection. For our prototype, we only focus on the azimuth angle. We assume people in the view of the camera would likely be on the sample relative plane as the camera, so the elevation angles would be negligible especially as the distance between the tag and the camera increases. Based on the camera's zoom level, we compare the azimuth with the camera view angle to determine if the tag is in the view of the camera.

The ContextCam also captures who is near or around the camera by simply storing the RF pings. This is important for detecting people who are at an event, but not the prime focus of the situation.

The biggest concern with our choice of an active tagging scheme is one of practical deployment. Active tagging enables effortless detailed annotation of people and objects in or around the camera. However, the problem with active tagging is the willingness of someone to actually distribute tags to individuals before capturing an event. Privacy problems also arise from an active tagging scheme. We can imagine that in the future each individual will wear or carry some sort of ubiquitous token. However, there are some motivating reasons why active tagging is a practical option for the ContextCam *today*. Researchers have already been able to successfully deployed large number of tracking or identification tags at events like conferences [5, 20].

Though "spur of the moment" situations would not work well with an active tagging scheme, most home videos depict settings or events that are known in advance, so "active tagging" one's whole family before recording is possible. The willingness to go through the trouble of distributing these tags comes down to the perceived cost-benefit for the user. As we have indicated, annotating the presence of individuals is the most time-consuming task, especially for large events like a family reunion or graduation. Taking 10 minutes to register and distribute the active tags may be a much better option than spending 2 hours to annotate a one-hour video.

Additionally, the active tags can be integrated into existing personal services like a cellphone or a Child Guard (a device that detects when a child wanders too far).

Another concern with active tagging is privacy. The active tags can allow users to opt-out by simply turning the device off or removing it if they do not feel comfortable being tracked on or around the camera. The ContextCam can provide a mechanism for individuals to opt-out of particular scenes, which is not possible with traditional videography. Although the individuals are still resident in the scene, there is an explicit indication in video metadata about their desire to opt-out.

5 Higher-Level Point of Capture Annotation

The metadata produced by the ContextCam at the point of capture is relatively low-level, which implies that some of the information is not directly beneficial to the user. High-level semantic metadata, the information produced manually in our previous annotation system, has more relevant meaning to the user. A tradeoff emerges between these two levels. Automatically captured low-level metadata tends to be very accurate. On the other hand, high-level semantic information is valuable, but introduces potential errors when automatically inferred. With manual annotation, the opposite is true. Whereas humans are good at identifying higher-level meaning and can do it rather quickly, detailed annotation of low-level information (time of day for a scene, when a person comes in and out of a scene) is particularly error prone. We show here how the low-level metadata from the point of capture can be translated and/or used to infer other higher-level semantic metadata. We also explore in the next section a way to help close this semantic gap with minimal human intervention.

5.1 Other Time Information

High-level time and date information is relatively straightforward to produce. Most common time and date semantics can simply be translated without any sophisticated inference scheme. For instance, the ContextCam determines holidays and seasons by simply looking at the timestamp and some calendar information. In this case the high-level semantic metadata is very accurate.

5.2 Inferring More Meaningful Location

The ContextCam produces raw latitude/longitude position coordinates for location information, which is indecipherable by the user. To extract more high-level meaning, we use a publicly available Geographic Information System (GIS) resource to determine the address. From this information, we retrieve street names, city, state, and sometimes landmarks. When used indoors, the ContextCam retrieves city and state information using the cellular network. And if we wanted to expend the effort, the same active tagging scheme could be used to tag indoor locations that could then be recognized automatically by the ContextCam. Cities and states are adequate for many high-level semantic tags, though there are even higher-level tags that may be

more appropriate such as "Disney World" or "Grand Canyon." Many of these higher level location tags can be inferred using the same technique we will show for inferring events.

The ContextCam's indoor/outdoor recognizer provides added information to help infer location and event based on other context information and past annotations. Though not directly valuable to the user, this information can play an important role in predicting annotations.

5.3 Inferring Event Information

Event information refers to what a particular video scene is about or *what* is being depicted within the scene, such as a birthday celebration, Christmas party, graduation, family reunion, basketball game, etc. Event information poses the most problems with automated annotation, since it typically consists of purely high-level semantic information that is almost impossible to detect directly with just sensors. However, event information can be inferred from the other context information and higher level semantics such as time, location, and people. For instance, we can infer a child's birthday party from the date, the fact that the child is of prime focus, and the people around, such as friends. All this information can be gathered with the ContextCam. The inference system becomes even more powerful when users validate or confirm the predicted event formation at the time of capture. Although event tags do not completely dominate the video annotation process, inferring and suggesting event tags can help alleviate some of the burden of manually searching and tagging from a large event tag hierarchy.

For the ContextCam we use a naïve Bayesian classifier to help infer higher-level event information based on the low-level context information gathered at the time of capture and verified past annotations. The information used for inferring includes the time, date, location attributes of a scene, and the people in and around the camera. Studies comparing classification algorithms have found the naïve Bayesian classifier to be comparable in performance with classification trees and neural network classifiers [17]. They also have exhibited high accuracy and speed when applied to large databases. The naïve Bayesian classifier is a statistical classifier. Given a scene and its current annotations, it can find the most probable high-level class that is consistent with the relative frequencies of past annotations and context attributes.

The root event node of the classifier represents the "class" of tags that the predictor tries to classify. A user-defined threshold determines the acceptance of inferred tags. Since the classifier determines the probabilities of all classes, there may be multiple tags that are inferred for a particular scene as long as the confidence is over the specified threshold.

The Bayesian classifier starts out with a minimal training set, but every time any scene is automatically annotated and appropriate event tags associated at the time of capture, a new entry is placed in the training set (stored as probability condition matrices on the camera). Thus the training set can grow fairly rapidly with just a few captured events. We tested the predictor using the case study archive mentioned above, which consists of 12 years (approx 12 hours) of video manually annotated with who, what, when, and where metadata. In the experiment, each test case represented one year from our case study archive, and the training set consisted of the one, two,

and three years preceding the tested year. We purposely chose one year chunks because most of the events in the archive were yearly (such as birthdays and reunions). Each year roughly included an equivalent number of who, what, when, and where (approximately 50 scenes per year and 400 tags in our case). Table 1 shows the prediction accuracy for the three cases. Our data suggests that one year of annotation data would be sufficient to produce fairly accurate predictions, requiring only three top suggestions to cover 95% of all cases. Taking more years (over 2 years in our case) into consideration actually hurts the accuracies, because family dynamics change over time. For example, family activities may change as a result of the addition of new family members, moving to a new home, and children growing up. However, an inference scheme that takes these potential changes into consideration so that more preceding years can be used in the predictor could be developed.

The accuracies shown in Table 1 do not take into account all the rich metadata that the ContextCam provides. Rather, it provides a baseline performance for simple user specified metadata. We suspect an actual long-term use (1-2 years) of the ContextCam will provide even better results, but we have not validated that claim.

Table 1: Accuracy of a naïve Bayesian classifier used to predict event tags. Each year consisted of approximately 50 user defined scenes with 400 metadata tags.

Number of directly preceding years used for training	Prediction accuracy	Number of event tags representing 95% confidence
1 year	79.6%	3.2
2 years	80.3%	2.8
3 years	80.1%	2.9

6 Storing the Metadata: Embedding Metadata into Video Frames

Storage of both low-level and high-level metadata to video scenes presents some interesting challenges with storage synchronization. MPEG-7 has emerged as a multimedia content description standard. However, MPEG-7 does not specify any standard for automatic AV description extraction or attaching of metadata to actual AV content. Commercial DV tapes allow minimal storage of metadata information like timestamps, but capture software rarely preserves this information and there is no clear standard that DV camera manufacturers follow. More importantly, these metadata storage units typically hold very limited information so they do not have the space to store a lot of metadata. Another approach is to store the information on a separate medium, such as another DV tape or compact flash, but this presents problems with synchronization. Now the user is burdened with two physical artifacts that add more complexity to the system when importing the videos into an archive or video editing tool.

Our solution is to multiplex the metadata directly into the raw uncompressed video captured from a digital video camera. We accomplish this by replacing every 60th frame (every two seconds when capturing at 30 fps) with a binary encoding of the metadata into the frame. So, at the 640 X 480 resolution of the digital camcorder in our prototype, there is 921 KB of space available for two seconds of metadata. When the video is extracted from the camera, the metadata is decoded from the raw video and the metadata frames are replaced with the previous frame. We found that even if the frame is not removed, there is very little perceptible difference between the encoded and non-encoded video. However, a problem arises when the video is compressed. Because the metadata frame is likely to be very different from the rest of the video frames (as a result of the encoded data), the video does not compress well. Replacing the metadata frame with the previous frame before compression alleviates this problem.

We also explored a simple steganographic technique that does not require replacing the metadata frame and encodes the metadata information in the least significant bits (LSBs) of the whole video frame. In other words the data is encoded using the last bits of the RGB values for each pixel. This scheme preserves the content of the video frame and is still visually similar to the original frame. This frame has a negligible impact on video compression and almost no perceptible difference during video playback (see figure 4). The tradeoff with this scheme is that at 640 X 480 the potential storage space decreases to 115 KB for 1 LSB, which is still enough space in our experience. Figure 4 shows the difference in picture quality for the metadata frame using more LSBs for increased storage. The metadata stays resident until compressed to another format. Although not used in the ContextCam, a compression resistant steganographic encoding technique [6] could have also been used to preserve the metadata through compression.

7 Interacting with the ContextCam

Having presented the ContextCam, we now explore the user experience of interacting with a point of capture annotation system. The ContextCam provides automated real-time point of capture annotation. Attributes like time, location, and people are attached to video with no human intervention, freeing the user to capture and enjoy the moment. Similarly, point of capture annotation also frees much of the burden of annotation after capturing the video. Browsing, filtering, and authoring capabilities of our video archive system briefly described in Section 2 are available immediately after recording.

Although our prototype ContextCam is a bulky modified laptop computer, the sensing hardware could be added to a commercial video camcorder with relatively little added bulk. The computational capabilities to support the active tagging triangulation and Bayesian inference engine are not computationally expensive.

Figure 4: An example of metadata encoded in each 8 bit RBG value for each pixel. Top left: Original Picture. Top right: Metadata encoded in 1 LSB. Bottom left: Metedata encoded in 2 LSBs. Bottom right: Metadata encoded in 3 LSBs.

What remains to be discovered is how the user would interact with the inference of high-level semantic metadata in order to accept or correct predictions while recording. Many digital video cameras already feature some sort of simple thumb or finger control (jog dial or arrow pad) to navigate a simple menu structure. Figure 5 is a screenshot of the ContextCam's prototype interface. The viewfinder occupies the center of the screen. The top portion of the screen shows the current event inferred by the real-time predictor. A user has two options for the event predictor. One is to set a threshold for an event to automatically associate with a scene. The other is to select the event from a list, ordered by likelihood, displayed near the upper right portion of the screen. After a user selects an event it is placed on the upper left portion of the screen. When the ContextCam suspects an event change, the currently selected event begins to flash. The user can either ignore the cue or select another event from the list. From our experience we found that two levels of event tagging were sufficient: event and sub event. A user can modify the sub event by pulling up the sub event list (same choices as the event list) and selecting the appropriate event. The ContextCam only uses one level when set to automatically assign event tags.

The ContextCam (see Figure 2) is equipped with a short-range, noise-canceling microphone on the back of the camera that is designed to capture audio commentary from the videographer. Although not currently implemented, we can imagine using the voice input and keyword spotting to test against the list of all previously entered events, providing further suggestions to the predictor. A relatively small dictionary of

current metadata tags in the family archive makes this a feasible real-time solution with existing speech technology.

Similar in spirit of NaviCam [25, 26], the ContextCam can provide an augmented interaction experience by overlaying context information (who, when, where) on the screen while capturing a scene (see figure 6).

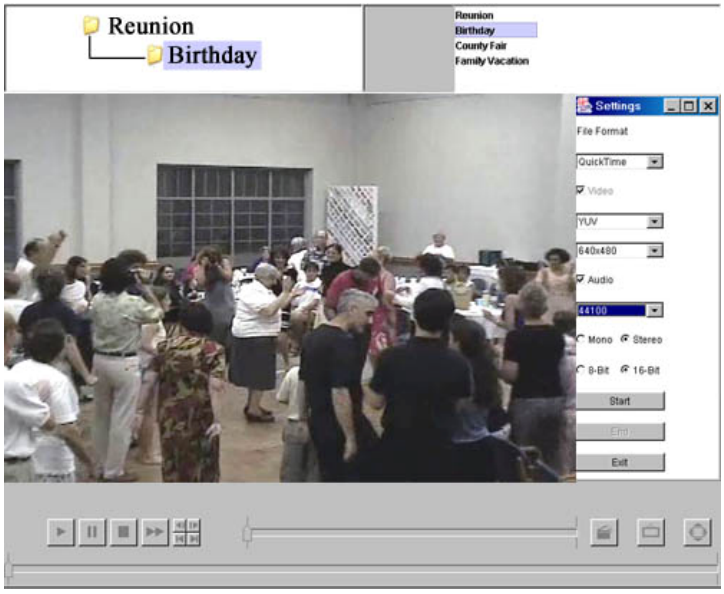

Figure 5: A screenshot of the interface on the ContextCam.

Figure 6: The ContextCam showing some live metadata (who and where) currently in the scene while capturing. The upper portion of the screen shows a thumbnail and name of each person currently in the view of the camera.

8 Conclusions

The ContextCam is a point of capture video annotation system that integrates a variety of sensing techniques to attach metadata of *when*, *where* and *who* directly to recorded video. It also uses simple mappings to infer high-level information of when and where, and a Bayesian predictor that uses point of capture metadata and past annotation history to suggest what event is currently being recorded. We presented the design of the ContextCam in detail and suggested how prediction capabilities and other features of audio annotation might be wrapped seamlessly into the video recording experience.

The original motivation of the ContextCam was to help reduce the arduous and time-consuming task of manual annotation of home videos, further amplifying the capabilities for filtering and sharing a family's recorded history. There may be other uses for point of capture annotation. For example, Su *et al.* suggest that video annotation at the point of capture using networks of wireless sensors might dramatically increase the capabilities for film production [28]. If we increase the accuracy of the ContextCam ultrasonic sensing by placing two more receivers on a vertical axis to determine angle of elevation, we could produce more 3-D tracking information, providing the potential for interesting visualizations of the recorded data. This would even allow for person identification after an event (*e.g.*, "Who is that person standing next to Grandma?") or even during the event (*e.g.*, when you lose a small child while on vacation). Most importantly, except for some issues with deploying tags and associating them to the ContextCam, all of the capabilities discussed in this paper are possible today without tremendous technology. For the visions of ubiquitous computing to become a reality, we need to see more examples of these kinds of capabilities made real in our everyday lives.

Acknowledgments

This work is sponsored in part by National Science Foundation (ITR grant 0121661). The authors thank Motorola, and in particular Joe Dvorak of iDEN Advancing Technologies Group, for the donation of the iDEN handsets used in this research.

References

1. Abowd, G.D., Gauger, M. and Lachenmann, A. The Family Video Archive: An annotation and browsing environment for home movies. *In the Proceedings of MIR*, November, 2003, Berkeley, CA.
2. Adcock, J., Cooper, M.D., Doherty, J., Foote, J., Girgensohn, A., and Wilcox, L. Managing digital memories with the FXPAL photo application. ACM Multimedia 2003: 598-599
3. Bederson, B.B. PhotoMesa: A Zoomable Image Browser using Quantum Treemaps and Bubblemaps. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2001)*, Orlando, FL, November 2001, pp. 71-80.

4. Casares, J., Myers, B.A, Long, C., Bhatnagar, R., Stevens, S.M, Dabbish, L., Yocum, D., and Corbett, A. Simplifying Video Editing Using Metadata. In *Proceedings of Designing Interactive Systems (DIS 2002)*, London, UK, June 2002. pp. 157-166.
5. Cox, D., Kindratenko, V., and Pointer, D. IntelliBadge: Towards Providing Location-Aware Value-Added Services at Academic Conferences, *UbiComp 2003*, Seattle, WA. Lecture Notes in Computer Science, 2003, vol. 2864, pp. 264-280.
6. Currie III, D.L., Irvine, C.E. Surmounting the Effects of Lossy Compression on Steganography, Proceedings of the 19th National Information System Security Conference, October 1996. 1996. pp. 194-201.
7. Davis, M. Media Streams: An Iconic Visual Language for Video Representation. *In Readings in Human-Computer Interaction: Toward the Year 2000*, eds. Ronald M. Baecker, Jonathan Grudin, William A. S. Buxton, and Saul Greenberg. 854-866. 2nd ed., San Francisco: Morgan Kaufmann Publishers, Inc., 1995.
8. Girgensohn, A., Boreczky, J., Chiu, P., Doherty, J., Foote, J., Golovchinsky, G., Uchihashi, S., and Wilcox, L. A Semiautomatic Approach to Home Video Editing. In *Proceedings of the ACM Symposium on Use rIinterface Software and Technology (UIST 2000)*, San Diego, CA, November 5-8 2000, pp 81-89.
9. Hakansson, M., Ljungblad, S., and Holmquist, L.E. Capturing the Invisible: Designing Context Aware Photography. *Proceedings of DUX 2003, Designing for User Experience*, ACM / AIGA, San Francisco, CA, June 5-7 2003.
10. Hazas, M., and Ward, A. A Novel Broadband Ultrasonic Location System. *In Proceedings of UbiComp 2002: Fourth International Conference on Ubiquitous Computing*, LNCS volume 2498, pages 264-280, Göteborg, Sweden, September 2002.
11. Jeon, J., Lavrenko, V. and Manmatha, R., "Automatic Image Annotation and Retrieval using Cross-Media Relevance Models," in the *Proceedings of SIGIR '03 Conference*, pp. 119-126.
12. Jiang, H., Helal, A., Elmagarmid, A., and Joshi, A. Scene Change Detection Techniques for Video Database Systems. ACM Multimedia Systems, 6:3, May 1998, pp. 186-195.
13. Kang, H. and Shneiderman, B. Visualization Methods for Personal Photo Collections: Browsing and Searching in the PhotoFinder. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2000)*, New York City, New York, August 2000, pp. 1539-1542.
14. Kender, J.R., and Yeo, B.L. On the Structure and Analysis of Home Videos. *Proceedings of the Asian Conference on Computer Vision*, January 2000.
15. Kipp, M. Anvil video annotation system. http://www.dfki.de/~kipp/anvil/. Page downloaded on August 1, 2003.
16. Kuchinsky, A., Pering, C., Creech, M., Freeze, D., Serra, B., and Gwizdka, J. FotoFile: A Consumer Multimedia Organization and Retrieval System. In Proceedings of the Conference on Human factors in computing systems (CHI 99). Pittsburgh, Pennsylvania, USA, May 15-20 1999, pp. 496-503.
17. Langley, P., Iba, W., and Thompson, K. (1992). An analysis of Bayesian classifiers. Proceedings of the Tenth National Conference on Artificial Intelligence (pp. 223--228). San Jose, CA: AAAI Press.
18. Lavrenko, V., Feng, S.L. and Manmatha, R., "Statistical Models for Automatic Video Annotation and Retrieval," submitted to the *International Conference on Acoustics, Speech and Signal Processing,(ICASSP)*, Montreal, QC, Canada, May 17-21, 2004.
19. Luo, J., and Savakis, A. "Indoor vs. Outdoor Classification of Consumer Photographs," Int. Conf. Image Proc. ICIP'01, Thessaloniki, Greece, Oct. 2001.
20. McCarthy, J.F., Nguyen, D.H., Rashid, A.M., and Soroczak, S. Proactive Displays & The Experience UbiComp Project. UbiComp 2003, Adjunct Proceedings, 12-15 October 2003, Seattle, WA.

21. Platt, J. C., Czerwinskim, M., and Field, B. PhotoTOC: Automatic Clustering for Browsing Personal Photographs. Microsoft Research Technical Report MSR-TR-2002-17, 2002.
22. Priyantha, N, Chakraborty, A., and Balakrishnan, H. The Cricket location-support system. In *Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking*, Boston, MA, August 2000. ACM Press.
23. Priyantha, N, Miu, Balakrishman, H., and Teller, S. The Cricket Compass for Context-Aware Mobile Applications. *In Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 2000)*.
24. Ramos, G. and Balakrishnan, R. Fluid Interaction Techniques for the Control and Annotation of Digital Video. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2003).* Vancouver, Canada, November 2-5, 2003.
25. Rekimoto, J and Katashi, N.: The World through the Computer: Computer Augmented Interaction with Real World Environments, *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '95),* ACM Press, pp.29-36, Pittsburgh, PA.
26. Rekimoto, J. NaviCam: A Magnifying Glass Approach to Augmented Reality, *Presence: Teleoperator and Virtual Environments*, Vol. 6, No. 4, pp. 399-412, August 1997.
27. Sarvas, R., Herrarte, E., Wilhelm, A., and Davis, M. Metadata Creation System for Mobile Images. *In Proceedings of the Second International Conference on Mobile Systems, Applications, and Services (MobiSys2004)* in Boston, Massachusetts. ACM Press, June 2004.
28. Su, N.M., Park, H., Bostrom, E., Burke, J., Srivastava, M.B, and Estrin, D. Augmenting film and video footage with sensor data. In *Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom 2004)*, Orlando, FL, March 2004, pp. 3-12.
29. Virage, Inc.. VideoLogger product. http://www.virage.com/solutions/details.cfm?solutionID=5&categoryID=1&products=0. Page downloaded on February 21, 2004.
30. Wactlar, H. D., Christel, M., Gong, Y., and Hauptmann, A. Lessons Learned from the Creation and Deployment of a Terabyte Digital Video Library. IEEE Computer 32(2), 1999, pp. 66-73.
31. Want, R., Hopper, A., Falcao, V., and Gibbons, J. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91-102, Jan 1992.