Synchronous Capture of Image Sequences from Multiple Cameras

P. J. Narayanan, Peter Rander, Takeo Kanade

CMU-RI-TR-95-25

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

December 1995

Abstract

Several applications today need to digitally capture every frame of a video stream or a camera. These range from psychological studies to surveillance to video processing. Some applications also need to capture the frames from multiple video streams synchronously and to correlate them with one another. A video stream of color data, though, represents a sustained bandwidth of about 26 Mbytes per second from the digitizing hardware to a secondary storage device without compression. This rate is well beyond the capabilities of most affordable systems today.

We present a system that can synchronously capture every frame -- or any user-specified subset of them -- from multiple cameras at full resolution and store them on a regular secondary storage device. The outputs of the cameras are recorded on tape using conventional VCRs. The Vertical Interval Time Code (VITC) is inserted onto each stream before recording. Each tape is played back, repeatedly under computer control if necessary, off-line on an editing VCR to grab the frames using a commercial digitizer. The VITC data is used to directly identify the frames while the tape is played back. We believe this is the first system in the world that can capture every frame from multiple video streams synchronously, fully scalable in the number of streams and the duration of capture. Finally, the system is inexpensive, costing $500 per channel. We present the system, its components, and the process of verifying the capturing process. We then discuss a few computer vision research projects made feasible by such a system.

1 Introduction

Several applications today need to digitally capture every frame of a video stream from a camera. These range from image/video processing to psychological studies to surveillance. Many also need to capture the frames of multiple video streams synchronously and correlate the data from multiple viewing angles. A video stream from a camera following the NTSC standards produces 30 frames of video data per second, each of which is typically digitized into 480 rows of 640 pixels with gray level values represented using an 8-bit number for monochrome images and using three 8-bit values for color images. Thus, a video stream represents a sustained bandwidth of 26 MBytes per second in color (9 MBytes per second for monochrome) without compression. Though most frame grabbers digitize in real time, this rate is well beyond the throughput of most secondary storage devices where the captured frames are to be stored, even with the best loss-less compression. Most image processing applications -- such as the stereo computation needed in our application -- cannot tolerate lossy compression, being interested in the minute variations of image structure.

There are a few commercially available high-end systems with the capacity to grab frames at video rate. They use expensive hardware and specialized disk systems to achieve the necessary bandwidth and throughput. For example a system containing digitizers and a set of the MD1 family of digital image recorders from Datacube, configured to give approximately 10 minutes of recording time, costs approximately $25,000 per video stream with additional streams costing nearly as much. An off-line solution to the problem can be achieved by recording each video stream onto a laser video-disc recorder after embedding its frames with unique time stamps. The high-end laser disc players can provide frame-accurate readout of each video stream while digitizing them off-line. However, such a recorder typically costs $15,000. A multi-stream system can get prohibitively expensive since each stream requires a separate recorder. Moreover, the system is designed for visual reproduction and could employ a lossy compression prior to storing that might have negative side effects in image processing applications.

Frankel and Webb developed a scalable multi-camera interactive video capture system using an iWarp, a general purpose massively parallel processor [1]. The digitized camera streams are fed to the internal data pathway of the iWarp and stored in the local memory modules of the node processors. The primary memory capacity of the iWarp limited the number of frames that could be held in memory. An iWarp system with 256 MBytes of memory can capture about 33 seconds (1024 frames) of a single monochrome video stream data or 2 seconds (32 frames) of 16 video streams. This system serves applications for which the above rates are sufficient and could be economical if an iWarp were already available.

We present a system that can capture every frame -- or any user-specified subset of them -- of many cameras at full resolution and store them on a regular secondary storage device. It stores the image data from each camera on an ordinary S-VHS (could use any format) tape. The Vertical Interval Time Code (VITC) is inserted onto each stream before recording. These tapes are played back off-line on an editing VCR and digitized using an off-the-shelf digitizer attached to a workstation. Each tape is played back on the VCR repeatedly under computer control until all the necessary frames are captured. The VITC is used to directly identify the frames as the tape is played. The time code is also used to correlate frames from multiple cameras which are all synchronized to a common sync signal. In order to digitize each video tape automatically, we use a computer-controllable VCR, although a manually-controlled VCR can be used for interactive digitization.

The strong points of our system are long recording capacity (limited only by tape length) and low cost per channel. The cost of the recording setup -- a VCR plus the VITC unit -- is $500 per channel, scalable to any number of channels. The digitizing setup for automated image capture costs $5000 for the Panasonic DS-850 VCR in addition to the cost of a commercial frame grabber. The weak points of our system are the following. One, the video data is stored on conventional tapes using commercial VCRs before digitizing, re-

ducing the visual quality. Two, the process could be time consuming for large numbers of video streams since each tape needs to be digitized independently and separately.

In this paper, we present how we record the outputs of the cameras synchronously and how we digitize the tapes to recover the synchronized video streams. The VITC time code plays an important role in our system. We, therefore, briefly describe the VITC standards in the next section. Section 3 presents our recording set-up and Section 4 presents the digitizing process. We describe our procedure to independently verify the completeness and synchronization of the process in Section 5. In Section 6 we sketch how the system is used in a few computer vision research projects. Throughout the paper we use numbers specific to the NTSC standards (525 lines, 30 frames per second) but our system is not (conceptually) limited to NTSC.

2 Vertical Interval Time Code

The Society for Motion Pictures and Television Engineers (SMPTE) has defined two standards of *time codes* that uniquely identify the frames of a video stream: the Longitudinal Time Code (LTC) and the Vertical Interval Time Code (VITC)[6]. Both standards assign a number to each frame in the hours-minutes-seconds-frames format, while VITC also encodes the field[1] number and includes 8 bits for error detection (using a Cyclic Redundancy Check, or CRC). The two standards differ in how the code is stored. LTC, consisting of eighty bits of time code for each frame, is stored on an audio track simultaneously with the video data. VITC, consisting of ninety bits of time code for each field, is stored on two horizontal scan lines during the blanking period of each field. VITC encodes the 1 bits in the data as short bright streaks and the 0 bits as dark ones. For each field, all ninety bits are stored on one scan line and repeated on the scan line two away from it for redundancy. The VITC inserts two synchronization bits before each group of data instead of all at the end of the code as done by the LTC.

Recording the LTC on an audio track by fanning the audio signal out to every recording device is a convenient method to correlate the frames of multiple video streams that are electronically synchronized with one another. However, LTC time codes cannot be reliably read at slow speeds. VITC does not have that drawback, being recorded as video data in each field. The time code can be read properly even in still modes. The extra sync bits and the CRC bits make VITC a more reliable time code standard. Another feature of VITC makes it attractive from our point of view: the time code can be acquired as bright and dark streaks in by the frame grabber as part of the captured image. It can be interpreted directly while digitizing or at a later time from the stored image. LTC would require additional hardware to convert the audio signal into something the computer can interpret, which would require extra synchronization between the video and LTC signals.

A line of VITC data consists of nine groups of 10 bits, each containing 2 sync bits and 8 bits of data. 32 User Bits, preset by the user at start, identify the run and are repeated on each field unless the user changes them. The hours of the time code (ranging from 00 through 23) are stored as two decimal digits using 6 bits (2 for the tens digit + 4 for the units digit). Minutes and seconds (00 through 59) are each stored as two decimal digits using 7 bits (3 for the tens digit + 4 for the units). The frame number (00 through 29) is stored as two decimal digits using 6 bits (2 for the tens + 4 for the units). Singular flag bits identify other information such as whether the field is odd or even. The last group of 8 data bits provide a cyclic redundancy check to validate the data. Figure 1 shows the time code digitized as a series of bright and dark streaks at the top of a frame. Consecutive lines of time code data are from different fields of the same frame. The time code information is repeated in each field after a gap of one scan line. Each bit of the data, whether 1 or 0, has a fixed duration. The total time for the 90-bits of one time-code line is 50.286 microseconds.

1. NTSC video contains 30 frames, or images, per second, with each frame composed of two interlaced fields. Field 1 contains the odd lines of the frame, while field 2 contains the even lines.

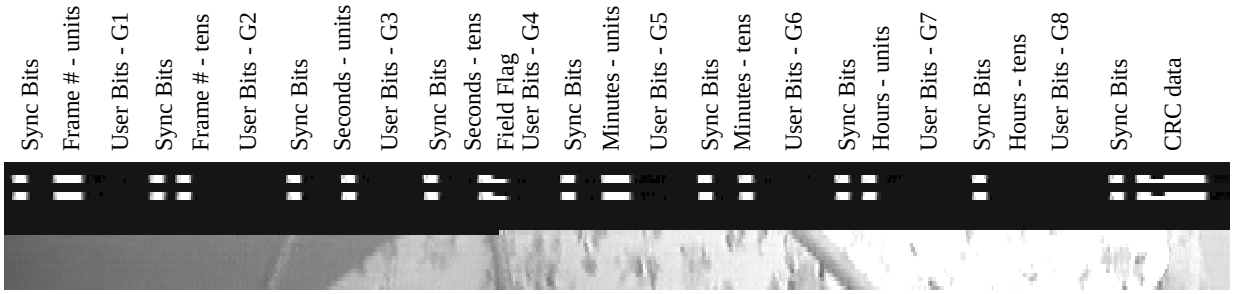

Figure 1: Vertical blanking portion of a frame containing VITC data

We make the VITC part of the digitized image by configuring the digitizer to grab the scan lines of the blanking period. This key feature enables us to grab all frames of the video stream reliably as explained in later sections. We extract the time code from the white and black streaks of encoded VITC data. Each bit of the time code spans approximately 7 digitized pixels[1]. The synchronization bits at the start of each group are used to identify the groups of time-code bits.

3 Synchronous Multi-stream Video Recording

Our approach to synchronously recording analog video from multiple cameras can be broken down into three steps: synchronize the video cameras themselves using a common sync signal so that they acquire image frames simultaneously, embed a time stamp within each frame in all video streams, and finally record the video streams to standard Video Cassette Recorders (VCRs). By synchronizing the individual cameras, the images in the video streams are acquired at the same “absolute” time. By recovering the unique time stamp embedded in every field of each video stream, we can easily correlate the frames from multiple streams. By storing the video for later use, we can perform slower, off-line digitization yet ensure the recovery of every frame of each stream.

A block diagram of our multi-camera synchronous recording system is shown in Figure 2. An external sync signal is supplied to each camera as well as to a time-code generator, synchronizing them to one another. The time code generator creates a unique time stamp in the Longitudinal Time Code (LTC) format, synchronized with the control signal. This time code is fed to each time-code inserter which inserts the VITC into the video stream. Finally, standard VCRs record the video stream in real time onto tapes.

4 Synchronous Multi-stream Digitization

A set of frames, one from each stream, recorded at the same time instant comprises a synchronous snap-shot of the multi-stream data. A number of such snap-shots, 30 in the NTSC system, are taken each second. The task of synchronous multi-stream digitization is to provide all the snapshots the user requests. We can achieve this by synchronously processing the tapes, recorded as described in the previous section.

It is common in image processing to digitize images from video tapes playing in a VCR. The user pauses the VCR at each of the required frames and instructs the computer to grab the frame.(The process of pausing and grabbing the required frames given a manually selected starting point could be automated using a computer controlled VCR.) This method has two drawbacks from our point of view. One, we have found that

1. An NTSC video frame is typically digitized to have 640 pixels horizontally. Since there are 90 VITC bits in each scan line, we get a bit time of $640/90 \approx 7$ per pixel.

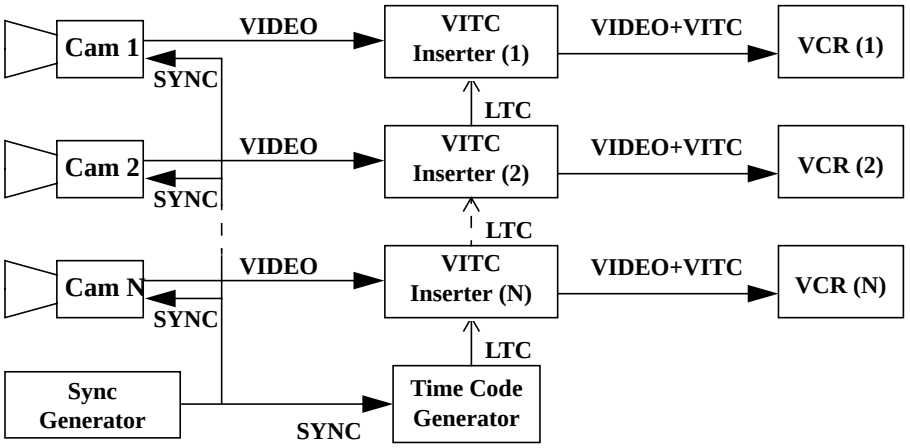

Figure 2: Block diagram of the synchronous, multi-channel, video recording system.

most editing VCRs are not frame-accurate. The frame number they think they are stopped at may be off by one or two frames from the real one, even on sophisticated VCRs. This makes it impossible to accurately correlate frames from two different tapes using the VCR to select time codes. The second reason concerns the resolution of the image output by the VCR in freeze or pause mode. Video "consumer" devices such as a monitor or a frame grabber combine alternate fields supplied to it into a frame. VCRs output the odd and even fields stored on tape alternately when playing, ensuring proper frame composition. Most VCRs, however, position the head over a single field when paused. That field is therefore output as both odd and even fields of the video data, reducing the effective vertical resolution by a factor of two. Figure 3 shows the drop in the vertical resolution on a portion of the image digitized in play speed and while paused. Finally, most VCRs do not allow the user to select the field on which to freeze, making it impossible to capture every frame of even one video stream in full resolution.

We overcome the frame inaccuracy by directly interpreting the time code from the image. We capture in full resolution by digitizing when the VCR is playing at the normal speed. A block diagram of the setup is shown in Figure 4. We configure the frame grabber — a commercially available K^2T V300 digitizer for Sbus developed at CMU — to grab the blanking period containing the VITC as part of the image. Figure 1 shows the VITC portion of a frame grabbed by the system. The black and white streaks of VITC data are interpreted by the computer on the fly to obtain the time code for each field as the tape is playing. If the current frame/field is one the user needs, the digitizer is instructed to freeze it so that it can be transferred from the digitizer's memory to the processor memory and/or the secondary storage. The tape continues to play and the system captures as many unread frames from the user's list as possible. When the tape goes beyond the last frame of interest to the user, it is rewound to the starting frame of the user's interest under computer control. This continues until all of the desired frames/fields are captured.

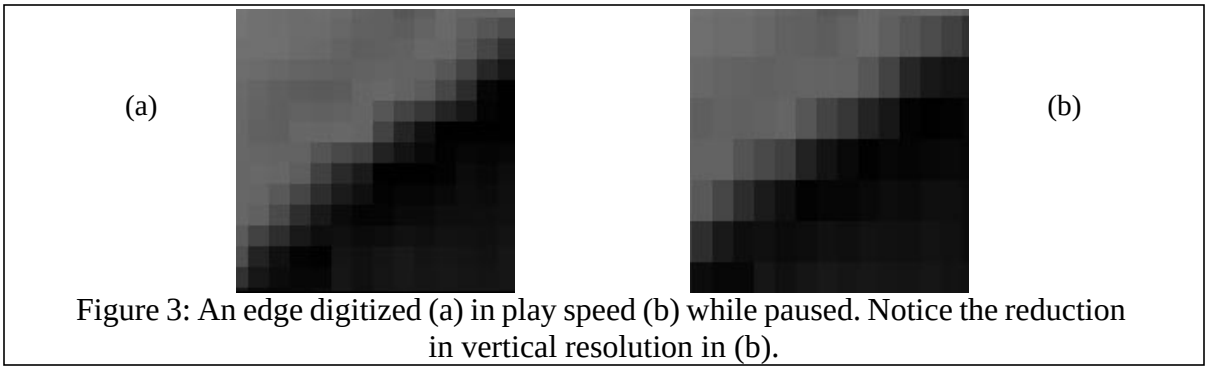

Figure 3: An edge digitized (a) in play speed (b) while paused. Notice the reduction in vertical resolution in (b).

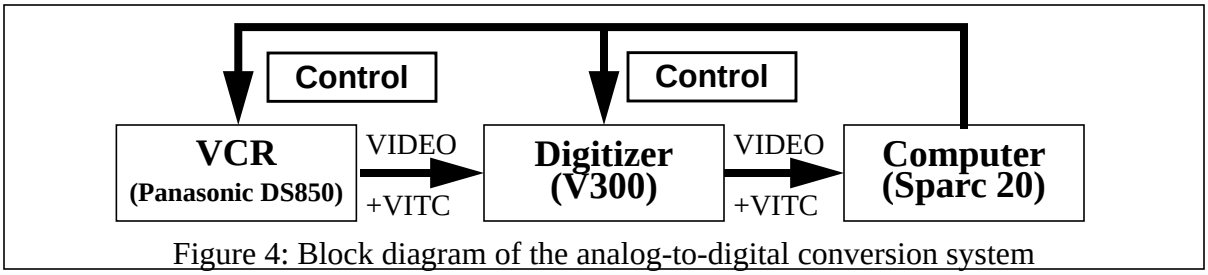

Figure 4: Block diagram of the analog-to-digital conversion system

The ability to grab a specific frame by directly interpreting the time code from the image reduces the bandwidth requirement from 26 Mbytes per second per channel to whatever the system can support. The lower the bandwidth the higher the number of passes required to get all necessary frames.

5 Process Verification

In order to guarantee the proper synchronization of the cameras, we must have a method of comparing the cameras to one another independent of the time codes inserted in their video streams. We accomplish this task by pointing the cameras at the display of a custom-built counter that is synchronized to the Sync Generator of Figure 2. This device counts the number of fields in the sync signal, then displays the number converted into seconds (up to 59), frames (up to 29), and fields (0 or 1, represented by an LED either on or off). Because the counter display changes synchronously with the master sync for the recording system, each field of video in each stream should record only a single state of the counter. If the cameras and the counter are not synchronized, they will record a combination of two states, so one test is to confirm that a single state is recorded in each field. The state in one video stream can also be compared to the state recorded in the other streams. If the systems are fully synchronized, then the counter display will be the same for all cameras, given a VITC time code to compare. Another way to perform this cross-camera check is to compare the difference between the VITC and the counter in one video stream to the difference in the others. Again, in a fully synchronized system, this offset should remain constant -- this is our second test.

The first test above implicitly assumes that each field of the video stream represents the information accumulated from no more than one field time (1/60th second in NTSC). Real cameras, however, frequently have two modes of accumulation, frame and field. Frame accumulation mode (or just Frame mode) effectively keeps the camera's shutter open for two consecutive fields (one frame), while field accumulation mode (or just Field mode, the one implicitly assumed above) keeps the shutter open only for one field. If the camera is in Frame mode, then the field no longer contains a single state of the counter, but now contains the combination of two consecutive states. This condition results in two troublesome phenomena. First, recall that the counter displays the field by turning on and off a single LED, changing state once every field. The accumulation of two consecutive fields in Frame mode results in the LED appearing to be on all the time, so we no longer have field accuracy in the observations of our counter. Second, once every frame the counter will increment the displayed numbers representing the number of frames. One field out of every two, then, will see the counter state before and after the transition, resulting in an image combining the views of the two states. Fortunately, the second phenomena is much easier to deal with, so we did not need a solution for it. The first problem, however, must be addressed to be able to guarantee field-accurate synchronization.

In Frame mode the shutter is open for a full frame (1/30th second in NTSC). In order to see an LED flashing, it should be off for at least that long during each cycle. By turning the LED on or off once each frame (half frame rate) we satisfy this condition. The physical sequence of the light, then, is ON-ON-OFF-OFF, with each step representing one field. The image sequence of the LED has the sequence ON-ON-ON-OFF, since

the LED will appear off only when it is physically off for the previous two fields. As a result, a single LED is insufficient for unique field identification in Frame mode. Fortunately, we can achieve this goal by using 2 LEDs, with exactly one on at a time. The resulting image sequence pattern in (ON,ON)-(ON,OFF)-(ON,ON)-(OFF,ON). With this pattern, we can identify Field 1 by both LEDs on and Field 2 by only 1 LED on.

The actual counter, then, contains 4 large 7-segment digit displays to show the seconds and the frames, 2 LEDs flashing at half frame rate each and with exactly one turned on at a time, and 1 LED flashing at frame rate, as shown in Figure 5. Figure 6 shows the timing of the counter display for the LEDs and of the images of the LEDs collected in Field mode. LED_1 is flashing at frame rate -- on for one field and off for the next. The image of LED_1 is represented by I-LED_1, which can be used to distinguish the fields: on during one field, off during the other field. Figure 7 shows the timing of the counter display for the LEDs and of the images of the LEDs collected in Frame mode. I-LED_1, again the image of LED_1 flashing at frame rate, is seen to be on all the time, even though the LED is off during half the fields. LED_2 and LED_3 are flashing at half frame rate, with exactly one on at any time, and their images are shown as I-LED_2 and I-LED_3. Note the 4-step repeating pattern: on-on, on-off, on-on, off-on.

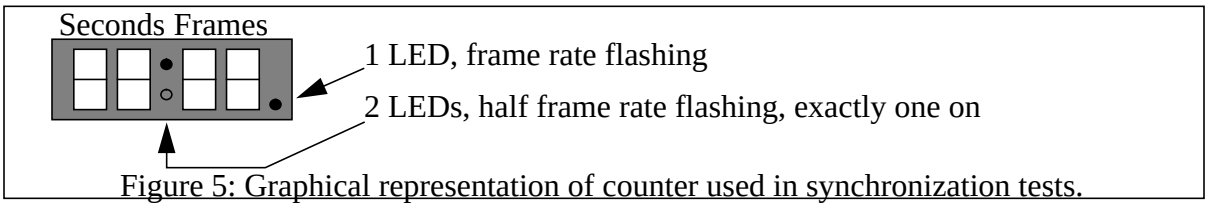

Figure 5: Graphical representation of counter used in synchronization tests.

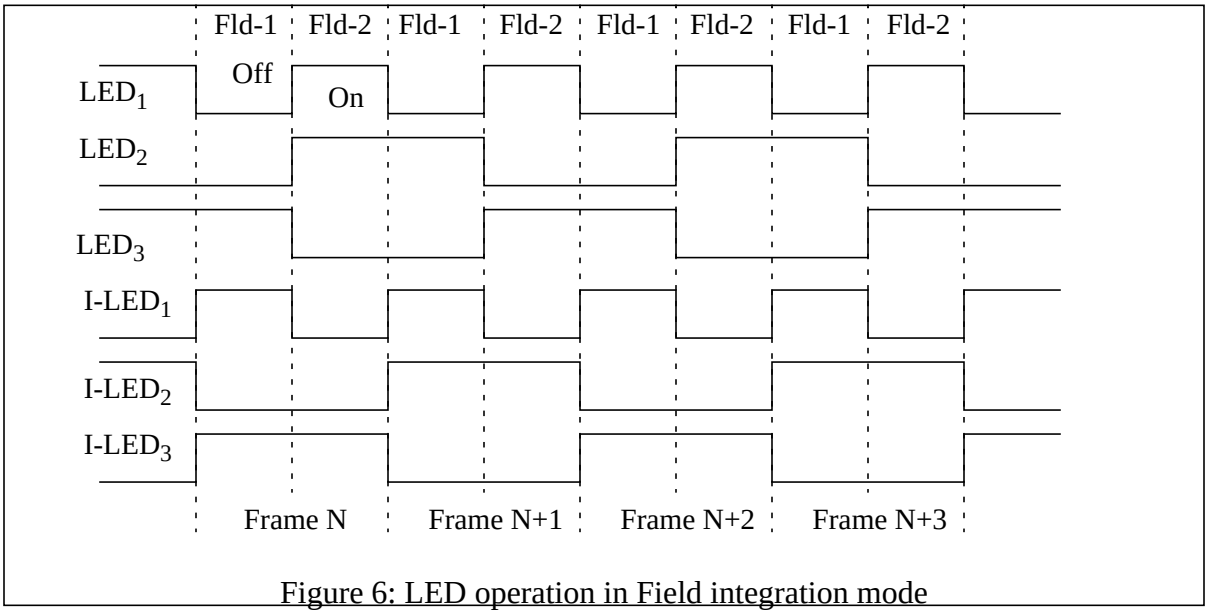

Figure 6: LED operation in Field integration mode

Figure 8 contains 4 consecutive fields (moving left to right in the figure) from each of two cameras watching the synchronization counter. The images in the top row of the figure come from a camera in Field mode, while the bottom images come from a camera in Frame mode. Each image contains two numbers, the 4-digit number from the counter as described above and an 8-digit number, beginning with the letter T, representing the interpreted VITC time code. For the Field mode camera, the only difference between the fields of the same frame is the frame rate LED, which is on for Field 1 and off for Field 2. In the Frame mode camera, two differences are apparent. As desired, Field 1 has both of the half frame rate LEDs on, while

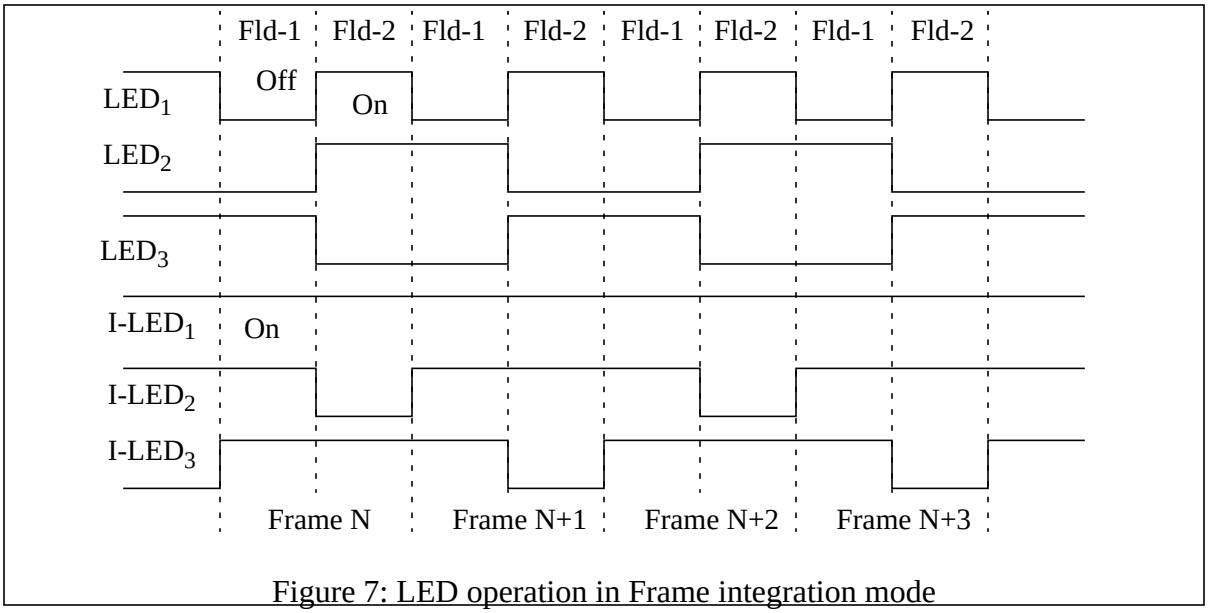

Figure 7: LED operation in Frame integration mode

Field 2 has only a single 15-Hz LED on. In addition, the numbers in the 7-segment displays are blurred during Field 1 because that field captures the transition between numbers. For the first image in the lower row, the transition is between 0928 and 0929. For the third image in that row, the transition is one of the worst: 0929 to 1000, which blurs all 4 digits. In both Field 2 images, though, the numbers are stable and therefore can be read correctly, allowing accurate determination of the synchronization. In this case, the system is fully synchronized.

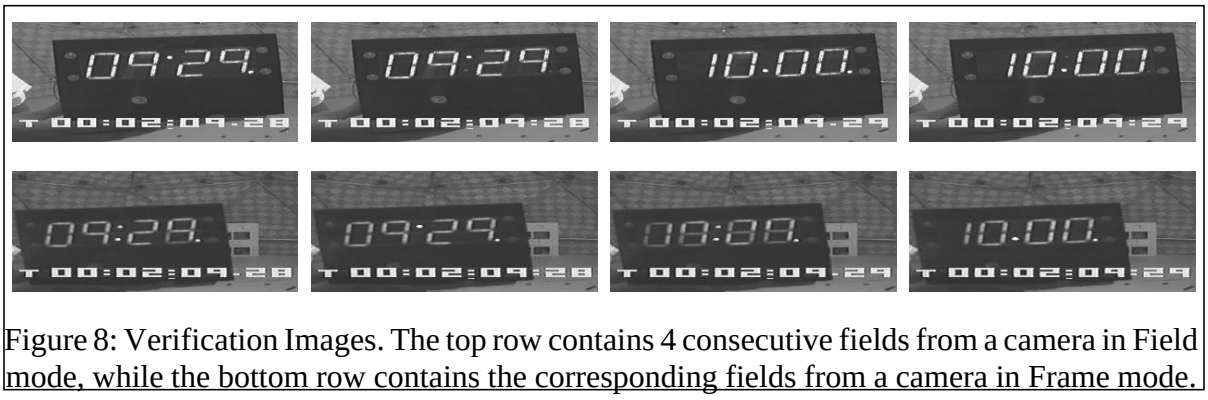

Figure 8: Verification Images. The top row contains 4 consecutive fields from a camera in Field mode, while the bottom row contains the corresponding fields from a camera in Frame mode.

In an actual cross-camera synchronization test, we do not require simultaneous views of the counter, but rather we compare its seconds and frames count to the seconds and frames of the VITC. The offset should remain constant so long as the system is fully synchronized, so our test is to compare this offset across all cameras. Since the self-synchronization of the camera -- that is, video to VITC -- has already been completed in the first test, this type of test is sufficient to guarantee synchronization. Figure 9 shows single fields from three different cameras during the same synchronization test. In this example, the counter is always 1 frame advanced relative to the VITC, verifying system synchronization.

Figure 9: Images used to validate system synchronization.

6 Applications as an Enabling Technology

In certain domains, applying our system does little more than ease constraints such as short recording length. More interesting, though, are the applications that were effectively infeasible without the technology represented by our system. One such application is Virtualized Reality (V-ized Reality) [3][4]. The fundamental goal of V-ized Reality is to immerse the user in a full 3D visual reconstruction of real events. Imagine, for example, being able to watch the NBA championships from any seat in the arena -- or even from the court itself -- or a medical student observing a V-ized Reality replay of the latest surgical technique to repair damaged heart muscle. These are just two of the many potential applications of such a system.

V-ized Reality seeks to automatically construct models of the real world that allow accurate reconstruction of any view, even views for which no physical camera exists. Virtual Reality (VR), on the other hand, uses hand-made artificial worlds that rarely bare much resemblance to the real world. In both cases, the models of the world are used to construct synthetic images from a “soft” camera whose position is controlled by the user. Because V-ized Reality relies on real images for reconstruction, the visual realism in the synthetic images is far superior to those in VR systems. In addition, V-ized Reality easily processes dynamic scenes, producing natural, consistent views of objects in motion. VR, on the other hand, has great difficulty accurately portraying motion because of the need for the dynamic models of the moving objects.

To achieve the accurate visual reproduction of real scenes, V-ized Reality uses a large number of synchronized image sequences of the scenes. Given a single time instant, the system reconstructs the shape of the scene by applying a multi-baseline stereo algorithm [5]. With the image sequences and with the computed scene structure, V-ized Reality uses simple computer graphics techniques to generate synthetic views of the scene. The scene structure is rendered as a 3D triangle mesh, while the images are texture mapped on the mesh to give visual realism to the rendered scenes. Having many views of the scene allows the system to more accurately reproduce the realistic interaction of soft camera position with scene structure and lighting.

In order to work properly, the V-ized Reality system requires synchronized image sequences sampled up to frame rate for a large number of cameras. The stereo computation requires the synchronized images of the scene. Without proper synchronization, the correspondences among the images are meaningless, since the structure of the scene may have changed between the samples. Second, dynamic scenes at times require acquisition of every frame in the video streams to accurately capture the motion. With too few samples, scene motion appears more like a collection of random images than like the dynamic scene from which they came. Finally, the realism of V-ized Reality images increases as the number of cameras increases, so it is desirable to have a large number of cameras. Without enough cameras, much of the scene may be occluded, potentially leaving large gaps in the scene models.

The Virtualizing Studio now contains 51 cameras mounted on a dome and pointed inside the dome, as shown in Figure 10. As discussed earlier, other methods of synchronous image sequence capture do exist, but even at $15,000 per channel, the recording system alone would cost more than $750,000. While such a recording system was technically feasible, the cost was too high to be practical, especially for a research

project with no guarantee of success. The development of a low-cost alternative, then, allowed the V-ized Reality concept to come off the drawing board and into a real research lab.

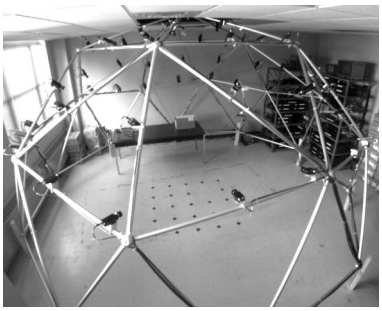

Figure 10: The V-ized Reality research platform, consisting of a dome with 51 cameras mounted on it and 51 VCRs that record the video from the cameras.

Several other research projects have also benefitted from our recording system. One project at the University of Maryland uses 3D models of humans to track and to recognize human motion [2]. They merge information from multiple views of dynamic scenes to find the 3D body pose at each time instant. This process requires synchronized image sequences and capture of every frame in the video streams in order to perform properly. A group at Carnegie Mellon University seeks to develop assembly plans for robots to follow by observing a human perform the same action. Again, the system needs dense sampling of synchronized video streams in order to correlate the information from multiple views. This group has replicated the recording system in their own lab, using the digitizing hardware in our system as needed -- an effective way to reduce costs in multi-project environments. A third project, jointly run at University of Pittsburgh and Carnegie Mellon University, is developing methods to recognize and analyze human facial expression. Multiple cameras provide information on the shape, size, and pose of the person's head in addition to multiple perspectives of the expressions. Without synchronization, the expressions in the images would not correlate, and may even conflict. This group has also duplicated the recording hardware in its lab and uses our digitizing system as needed.

7 Conclusions

We presented a novel and economical system to synchronously capture every user specified frame or field of an arbitrary number of video streams. Each stream is synchronized to a common signal and stored on a video tape with VITC time code inserted into each field. These tapes are digitized off-line, identifying the needed frames by interpreting the VITC time code on-line using a commercial frame grabber. The user-specified frames can be grabbed automatically using a computer-controlled VCR. The automatic digitizing system costs $7000 for the first video stream of which $5000 is for the VCR. A system that relies on manual control of the VCR could use a low-cost VCR like those in the recording system, or could even re-use one of the recording channels for playback during digitization. Additional recording channels cost $500, for a VCR and the VITC inserting equipment. Finally, the system in enabling research into applications considered infeasible without low-cost synchronized recording.

8 References

1 M. J. Frankel and J. A. Webb. Design, Implementation and Performance of a Scalable Multi-Camera Interactive Video Display System, *Proceedings of Computer Architectures for Machine Perception*, Como, Italy, September 1995.

2 D.M. Gavrila and L.S. Davis. 3-D Model-based Tracking of Human Upper Body Movement: a Multi-View Approach, *IEEE Symposium on Computer Vision*, Coral Gables, U.S.A., Nov 1995.

3 T. Kanade, P. J. Narayanan, and P. W. Rander. Virtual*ized* Reality: Concept and Early Results, *IEEE Workshop on the Representation of Visual Scenes*, Boston, June, 1995.

4 T. Kanade, P. J. Narayanan, and P. W. Rander. Virtual*ized* Reality: Being Mobile in a Visual Scene, *International Conference on Artificial Reality and Tele-Existence and Conference on VIrtual Reality Software and Technology*, Japan, Nov 1995.

5 M. Okutomi and T. Kanade. A multiple-baseline stereo, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353-363, 1993.

6 Society of Motion Picture and Television Engineers. American National Standard for Television -- Time and Control Code, *SMPTE Journal*, June 1986.