GCERSync and the Live-Blog Challenge: Preserving and Archiving the GCER Experience

Braden McDorman
Jeremy Rand
Matthew Thompson

Norman Advanced Botball braden@betabot.org
Norman Advanced Botball biolizard89@gmail.com
Nease High School matthewbot@gmail.com

GCERSync and the Live-Blog Challenge: Preserving and Archiving the GCER Experience

1 Introduction

When KIPR first formed the Botball Youth Advisory Council (YAC) for the 2009 season, one of the first things we proposed in the YAC was to work on methods of preserving and archiving the experience of GCER for everyone to see. Our logic was twofold: Botball teams could relive the excitement, and Botball and GCER would be exposed to millions of Internet users (including potential participants and sponsors). The two main projects we came up with to realize this goal were recording the game matches for online viewing, and encouraging teams to live-blog the conference. These projects evolved into GCERSync and the GCER Live-Blog Challenge, respectively. As with most things, solidifying our ideas would be much harder than postulating them. This paper is a presentation of the problems facing us, techniques used to overcome them, and our end products.

2 GCERSync

In the past, third parties have posted individual matches on YouTube. However, the vast majority of matches are never recorded and are thus lost forever. KIPR has stated that they do not have the resources to manually go through recordings of the game tables (a 3-day tournament with 5 hours of action per day and 8 game tables would yield 120 hours of video), so we needed a smarter way to get videos of the matches. We came up with GCERSync.

2.1 Recording

2.1.1 What to Record?

When GCERSync was first conceived, we had planned on capturing video streams from the game tables' camera, cameras on robots (we called the concept the Bot View), and all of the breakout sessions and speakers. We ended up scrapping the cameras on robots due to several problems. First, getting teams to consent to run custom software on their tournament

robots would be tricky, particularly since we are well-known for having the hacking expertise necessary to write well-hidden malicious code. Second, the quality of the Bot View would be poor at best, since the CBC's webcam driver is quite low-quality. Finally, we were concerned about eating CPU time on the robots. We hope to fix some of these issues and try a Bot View feature next year. Breakout sessions and speakers were also later scrapped, because of the cost of the necessary equipment (and the relative ease of manually cutting the video from a traditional digital video camera compared with manually cutting hundreds of 135-second Botball matches). Again, we hope to try again next year.

2.1.2 V4L2 Recording on CBC-side

Our first attempt was to record MJPEG compressed data from a Microdia webcam (the webcam in the Botball kits), and save to a USB mass storage device or to a network device using rsync. This idea would be very cheap (KIPR has lots of Microdias lying around), but we ran into problems. The webcam provided by KIPR is based internally on the microdia usb camera chipset. Its open source driver is unfortunately slow and buggy. This is why, for instance, the cameras resolution is stuck at 160 by 120 on the CBC. A resolution higher than that and the stream becomes corrupted. For the Bot View, where this was the only method possible, we were still hopeful we could overcome these shortcomings and provide at least a mediocre view from a robots perspective. Video for Linux 2[5] (V4L2) is a standardized interface to access cameras on Linux. To actually record the stream on the CBC, we planned to use the program ffmpeg in conjunction with the V4L2 stream /dev/video0. However, the large dependency list required to even compile ffmpeg for an ARM platform discouraged us even further, and we abandoned the idea entirely half way through the 2010 season when we found that the CBC Firmware takes exclusive control of the Microdia, and that we would need to do heavy firmware vision mods to allow video to be captured while vision was in use.

2.2 CBC/PC Sync

Our next goal was to capture a video stream from the table's camera. But how would we know when a match began, or ended? KIPR didn't have the resources to operate the program manually, and our original goal was complete autonomy. Then we realized that every table has a CBC that controls the game's progress. Our teams regularly use a network to communicate with CBCs, so it was a simple conclusion to apply the same technique here. We would have a server (PC) that listened for events from the table CBCs, and would begin and end recording based on that information. Internally, this is accomplished over a UDP multicast socket, which all CBCs send data to. The IP of every CBC is then stored in a list for later lookup. Our server side program needed to be easy to setup, so we wrote a Qt based application to save and manage CBC connections. In the future this program will also talk to KIPR's match making software, allowing us to know which teams we are viewing on each table.

2.3 Adding metadata with AviSynth

Once the video files are captured, it is necessary to add an intro and ending with metadata from the match (e.g. the team names and the final score). We opted to use AviSynth[1] for this, as it allows the process to be completely automated by command-line utilities. The intro contains the game number, each team's name and team code, which bracket number they are in, the teams' previous seed scores and their number of D.E. losses, and the teams' promotional slides which were sent to KIPR for GCER documentation. An overlay of some of this data is displayed in the main video as well. The ending lists the final score and where the teams' next match will be. This data is gathered from KIPR's bracket software. We hope that the resulting videos outputted from AviSynth will allow someone who has downloaded the video from YouTube with a cryptic filename to immediately recognize the context of the video.

2.4 3D Graphics with SolidWorks Animator

While AviSynth can do a lot, nothing screams "professional" like 3D-animated graphics. SolidWorks Animator[4] is a perfect choice for animating the board to produce a nice intro video. KIPR already provides the CAD model, and SolidWorks Animator supports custom camera motions and photorealistic rendering, so we hope that the result looks nice. To finish it off, we use AVISynth to place intro text and graphics on top of the game board, in a manner which we hope looks well-synchronized.

2.5 Upload to YouTube

With hundreds of YouTube videos to upload, doing it manually isn't an option. Luckily, Peteris Krumins has created a Perl script[8] which uploads any video to YouTube. With this tool, we can simply use a batch file or Bash script to iterate through the videos produced by AviSynth and upload them while we sit back. We also had concerns about the number of videos. The official YouTube API has a lifetime limit on the number of uploads an account holder can make with the API. Luckily, Peteris's script does not use the API; it simply emulates a human-operated web browser uploading the file, which alleviates this concern. Finally, the length of the videos was a question when we were trying to do the speakers. YouTube's limit is 10:59, but many breakout speakers are 15-20 minutes, and the featured speakers are often over an hour. We ended up discovering that KIPR has a Nonprofit account with YouTube, which has no such limit.

2.6 Project Delays

Due to lack of funding we were unable to purchase the required recording equipment for GCERSync. We hope to try again at next year's GCER.

2.7 GCERSync Web Site and Future Work

KIPR's bracket software produces web pages. In that spirit, we plan to produce a section of the Botball Community to house the videos. It will allow easy navigation of the videos, and facilitate following a team throughout the tournament. In future years, we would like to use the website to provide live in-browser video feeds of conference events (game matches as well as speakers) for HTML5-based browsers, as well as RSS notifications of conference events for specific teams which the user wants to follow. In addition, we would like to have the videos automatically made available via both YouTube and BitTorrent the moment the match is over and metadata is available (a human pressing a button or moving a flash drive would be unnecessary). Time will tell whether we can implement this in a timely manner.

3 GCER Live-Blog Challenge

Many teams (ourselves included) have been in frequent contact with friends during GCER, as well as the regional tournaments, telling them what's going on. At GCER 2008, enough of Jeremy's friends wanted to know what was going on that he simply didn't have the ability at an already time-constrained event to IM each of them with updates. Instead, he decided to set up a password-protected blog, and prior to the conference sent a link and the password to the blog to the friends who had expressed interest. The experiments performed by Jeremy during GCER 2008[6] were somewhat successful, but the Internet access at the conference died suddenly, causing the blog to end during the second seeding round. A KIPR official was somewhat interested in the blog idea, and after agreeing to try again next year, Jeremy made the blog publicly viewable after the conference. GCER 2009's blog[7] ended similarly to 2008's; lack of network access caused a lack of posts, although the blog did last until partway through Double Elimination. By this point, we had joined the YAC, and we decided that the blog concept, if successful, would be an excellent way to increase Botball's Internet exposure.

3.1 Format of the Blogs

We decided to keep the format of the blogs fairly free-form. Teams may need to post quick notes when they're pressed for time, but may be able to write a couple paragraphs at the end of the day; for this reason, we don't have requirements on post length. Teams can also choose the medium they wish to use for each post; text, pictures, audio, and video are all acceptable. Teams are encouraged to use whatever post length and medium works best for their specific situation and message. We feel that letting teams decide what's best will produce better quality and more compelling blogs than if we had strict formatting requirements ahead of time. We look forward to seeing what styles participating teams come up with, and we hope to see a good bit of variety between teams.

3.2 Dealing with Poor Network Access

We don't want teams to abandon their blogs when the network dies, as happened to Jeremy two years in a row. Because of this, we decided to recommend Posterous[9] for the blogs. Posterous is a blog host which allows posts, including pictures, audio, and video, to be sent by e-mail. Since nearly every mobile phone on the market can send texts, pictures, audio, and video to e-mail addresses, this effectively allows teams to post from a phone, even if they are nowhere near a laptop or if the network is down. For example, a team might post an audio recording summarizing their test run on the way back from the game table.

3.3 Who Has Time to Blog?

Obviously, no one at GCER will have time to blog every last detail as soon as it happens. However, the design of the Live-Blog Challenge minimizes this problem. First, most teams have multiple members present. Since at most times, at least some of the team members are likely free, teams can divide up the blogging based on who is busy. This is more likely this year than previously, since GCER 2010 has a documentation element, and the designated documenters may not be involved in programming or engineering. The ease of posting to Posterous via a mobile phone also lowers the time commitment required to post. Finally, many teams will be participating, and presumably different teams will be active at different times (while one team is at a game table, another may be in the pit, and another may be eliminated). This means that once the blogs are made public, a reader who reads multiple blogs will get more information than one who reads one blog – this is good, since not only is more information present, but the PR effect is increased by encouraging the reader to follow multiple teams.

3.4 Judging

Judging is difficult for this kind of competition for several reasons. First off, blog quality is difficult to quantify, so the judges are instructed to rank the blogs from best to worst, based on the blogs' "ability to effectively convey the experience of competing at GCER to someone who has never been to GCER, and may never have competed in Botball." Since some of the judges are YAC members whose teams may be competing in the Live-Blog Challenge, judges must not include their own school in their ranking. The judges' rankings will be combined, likely using the Condorcet[3] algorithm. Another judging issue is that since the blogs run through the end of GCER, it is impossible to announce winners at GCER. Instead, we will announce the winners after GCER concludes. We aim to have the judging completed within 3 weeks, but it may be much shorter, or possibly longer, depending on our post-GCER summer schedules and the number of participants.

3.5 Live-Blog Challenge Future Work

Posterous's ability to accept posts by e-mail gave us the idea of having a bot (named Botguy) post on teams' blogs via a scripted e-mail when interesting events occur relating to their team. For example, videos of the teams' game rounds created by GCERSync would be posted on the teams' blogs automatically, and get merged with the team members' posts about those game rounds. Similar automated posts would include information about scoring data, events such as GCER paper presentations, when teams are on deck, and awards won. Sadly, we did not have time to program this for GCER 2010. Keep an eye out for next year, though!

4 Conclusion

If you like these projects and want to see future work done on them, now's your chance to get involved. YAC applications for the 2011 season will be released soon on the Botball Community. If you have a bit of free time and want to help promote and improve Botball, we'd love to have you on the YAC. And even if you're not interested in joining YAC, post ideas on the Botball Community[2], and there's a good chance a YAC member will take them into consideration. From YAC, we wish you an awesome GCER!

References

[1] AviSynth. `http://avisynth.org/mediawiki/Main_Page/`, June 2010.

[2] Botball Youth Advisory Council. `http://community.botball.org/`, June 2010.

[3] Condorcet Method. `http://en.wikipedia.org/wiki/Condorcet_method`, June 2010.

[4] SolidWorks. `http://www.solidworks.com/`, June 2010.

[5] Gerd Knorr. video4linux. `http://linux.bytesex.org/v4l2/`, October 2005.

[6] Jeremy Rand. Norman High School: 2008 International Botball Tournament Blog. `http://nhsbotball2009.wordpress.com/`, July 2008.

[7] Jeremy Rand. Norman High School: 2009 International Botball Tournament Blog. `http://nhsbotball2009.wordpress.com/`, July 2009.

[8] Peteris Krumins. How to Upload YouTube Videos Programmatically. `http://www.catonmat.net/blog/how-to-upload-youtube-videos-programmatically/`, July 2007.

[9] Posterous. Posterous. `http://www.posterous.com/`, June 2010.