October 2, 2005

Votoscope Software by Harri Hursti

So, what I have been working on has now been named as "Votoscope". It is free, open source software - when I release it mid-month, it is once and for all out of my hands. As source code, it is not polished to be optimized, absolutely opposite, sacrifices in source code everything to be easy to understand with less than average programming skills - it is more a tutorial how to build a optical mark recognization software than a end product, but it gets the job done.

The idea was very simple :

1) I found out that high speed scanners of Diebold are actually 3 step process - first ballots are imaged as digital images and stord as individual files to image server and then images are interpreted by OMR software and then results are transmitted to cenrtal tabulator for final processing - but the files are left to image server. Other manufactures have similar design for their high speed scanners - hartintercivic confirmed that this is the case with all their scanners including precinct based.

2) Few election supervisors had made point to me that ballots are public records and citizens have access to them via Freedom on Information Act and Public Record Requests. As paper ballots those can not be allowed to be more than inspected to preserve intergrity. In case of digital image exists... in most cases only reasonable cost to copy it on CDs or DVDs. So, no new laws are required.

3) I got access to few 100s of images from number of states - I was shoked by poor quality of the images -- bad maintenance, bad calibration, auto-calibration going wrong, physical paper trail problems etc - things were seriously wrong even in square one. Then I learened (as fact I know, not hearsay) that some manufactures had put speed over accurancy and assumptions over user interventions to gain competitive edge.

So, I decided to write my own software to analyze the images - and then I also wrote a simple tabulator software with "what-if capabilities" and this means "what if instead of 18% of grey the sure recognition without user intervention would be 19%" and this is valid question, because locations running with auto calibration EVERY BALLOT has different calibration, even within the same ballot, BUT TRESHOLDS ARE FIXED IN COMMERCIAL SOFTWARES! This is actually, in my opinion, much more important than my previous discovery. Even this tutorial-level program processes one race at the time, with normal $700 home PC, over 10000 ballots per hour - and it is learning as it goes, which helps user to teach it ballot styles and rotations only as needed making even counties of 1000s ballot styles managable. It is also network enabled, so who ever having number of computers in local area network can pararrell process the election each workstation processing over 10000 ballots per hour and one user entering new ballot styles and rotations as needed. My own "central tablulator" imported 250000 ballots into database in less than 15 minutes (one race, same home pc) and processed with forensic information (like reference white balance from between ovals to track bad calibration) in less than 30 minutes.