BATCH PROCESSING OF REMOTE SENSING JOBS ON THE PC

Martin Ozga
National Agricultural Statistics Service
United States Department of Agriculture
3251 Old Lee Highway, Suite 305
Fairfax, VA 22030
mozga@nass.usda.gov

ABSTRACT

The National Agricultural Statistics Service of the United States Department of Agriculture estimates crop acreage for entire states or major portions of states using satellite data, mostly Landsat 5 and Landsat 7. For the 1999 crop season, Landsat processing was done for Arkansas, Illinois, Mississippi, New Mexico, and North Dakota using more than 50 scene locations, many multitemporal. A system has been developed using mostly locally developed and some commercial software to do batch processing on Windows NT PCs. The main procedures which are implemented for batch processing are scene reformat, multitemporal scene creation, collection and grouping of ground truth pixels followed by supervised clustering, and maximum likelihood classification of entire scenes, generally multitemporal and using all bands. The user submits jobs using a GUI interface. The submission program checks that all required files are available and, if so, submits the job. There is one PC which is the batch server which acts as a central repository of batch jobs. The jobs are actually run on any Windows NT PC on the network which has been enabled to run batch jobs. This allows the use not only of dedicated batch PCs, but also other PCs during nights and weekends.

INTRODUCTION

The National Agricultural Statistics Service (NASS) of the United States Department of Agriculture (USDA) has long done crop acreage estimates based on areas known as segments from an area sampling frame. These segments are visited by enumerators who interview the farm operators to get information on the covers and field sizes. Also, the segments are located on aerial photos which allows the segment boundaries to be drawn relative to the satellite data. The main use of the segments is to estimate total crop acreage for various crops using direct expansion statistical techniques on the sample segments. For quite some time, an ongoing research project at NASS has been to use the segment data to do training for classifying full satellite scenes over areas of interest. Then, a regression estimate is computed using both the segment data and the results of the full scale classification to get improved estimates. Also, the categorized scenes may be used to display areas where various crops are located. In many cases, multitemporal scenes, consisting generally of a spring and summer scene, are used to improve the cover separability.

Since the Landsat data has seven bands and the multitemporal data thus has fourteen bands, substantial processing is required for certain of the steps, especially for the full scene maximum likelihood classification. Previously, supercomputers and mainframes were used for these tasks. Since the NASS remote sensing project is quite small, we could not afford our own supercomputers or mainframes so they had to be used from remote locations on a timesharing basis. This was both cumbersome and expensive. PCs are now sufficiently fast and with enough memory and disk capacity so that it is possible to do satellite data processing on the PC. WINDOWS NT, while not yet as comprehensive as a mainframe operating system, provides many capabilities allowing development of a batch system, which while less sophisticated than those available on mainframes, is suitable for our purposes. Some consideration was given to using workstations, but PCs are now as powerful as all but the top of the line workstations and are significantly less expensive. Also, PCs are easier to export to the state offices which already have PC based local networks. Since PCs are typically part of a network, additional PCs may be acquired at a reasonable cost until the required throughput is attained. However, it is tedious for the user to repeatedly interactively execute long running applications on various PCs, so some sort of batch system is useful. Mainframes typically have batch capabilities for such long running jobs, but not PCs. Also, batch operation allows jobs to be queued up and run during nights and weekends without user intervention.

NASS uses the PEDITOR system, developed in house, for these remote sensing applications. This system was

originally developed for various computers, but has been converted to run on the PC under WINDOWS NT. First, a general description of the batch system is given and then the implementation of the various remote sensing jobs is described.

Some explanation of our terminology related to satellite data is useful here. A window file is a portion of a scene which may be the entire scene or just a small portion of a scene containing a single segment. A multiwindow file contains several windows, the chief use being to store the satellite data for several segments. A multiwindow file containing satellite data for all segments in a scene is considerably smaller than the entire scene, allowing storage space to be saved until the entire scene is required for full scene classification. With disk storage capacity expanding and prices dropping, the necessity for using multiwindow files decreases. A satellite scene is presented as a rectangular area. The actual satellite data often covers only a part of this rectangle, due to adjusting the scene to geographic coordinates. The area in the rectangle but with no actual data is referred to as filler.

The procedures done in batch mode are REFORMATTING SCENES, MULTITEMPORAL SCENE CREATION, PACK AND CLUSTER, and FULL SCENE CLASSIFY. REFORMATTING SCENES converts a scene from the format supplied by the vendor to the PEDITOR format. MULTITEMPORAL SCENE CREATION overlays two scenes from the same area. PACK AND CLUSTER assembles all the pixels for various covers and then does clustering. FULL SCENE CLASSIFICATION does a maximum likelihood classification of full scenes. Each of the procedures is discussed below, but first a brief discussion of the mechanics of the batch system relative to WINDOWS NT is in order.

BATCH PROCESSING USING WINDOWS NT

The programs to be run in batch are all character mode rather than GUI mode. Under WINDOWS NT, the character mode windows are referred to as consoles. Consoles may be made sufficiently large to retain all of the output of a program so that, after execution, the output may be saved to a log file for later review. This console capability is not available in WINDOWS 95 or WINDOWS 98, making them unsuitable for the type of batch processing done here. WINDOWS NT also can make use of PCs with two processors, quite useful in allowing more batch jobs to be run since many of the batch jobs are quite computationally intensive. We believe that these capabilities will also be available under WINDOWS 2000, but as of this writing, WINDOWS 2000 has not been available for experimentation. WINDOWS NT does not come with any built in batch capabilities.

Aside from WINDOWS NT, the other commercial software used is XLNT. XLNT is a scripting language derived from, but not identical to, the scripting language DCL used on the VAX/VMS operating system. XLNT has some important advantages over the MS-DOS scripting language standard under WINDOWS NT in that the same file may contain both program executions and data so that all input for a batch execution may be put into a single file rather than having separate files for the program executions and the data. Also, XLNT provides capabilities for conditionals, loops, and jumps. Thus PCs which run batch jobs must have XLNT. However, XLNT is not required on PCs which only submit jobs.

One PC is designated as the batch server and all jobs are submitted to that PC in a designated directory. This batch server need not be a network server and need not be running the server version of WINDOWS NT. The program BQUE, developed locally, runs on the batch server waiting for an interrupt due to a change in that designated directory. Such a change occurs when a job is submitted or when one of the batch queues has taken a job and executes it. If a job is submitted, the name of the job file is placed in a standard named file and BQUE adds that job to the end of its list and deletes the standard named file. If a job has begun execution, BQUE places the first job in its list out for execution. Jobs are sent for execution in the same order as received, no priority system is in effect.

The actual batch queues are simply XLNT scripts which are either running a job or else checking if a job is available to be run. When not running a job, these batch queues wait for a short time and then check for the standard named file containing the job file to be run. If found, this file is renamed, read to obtain the job file name, and deleted. This causes BQUE to be interrupted so that it can put out another job to run, if any are in the queue. This scheme allows several PCs to be running batch queues and also PCs with more than one processor to be running more than one queue. Thus, over nights and weekends, other PCs may have batch queues started and then discontinued (if not running a job) when they might be used for other tasks.

The BATCH program, developed locally, provides a GUI interface for submitting jobs. The user enters various parameters depending on the type of job, as described below in the discussion of the various jobs. If all required files are available, the job is submitted, otherwise the user is advised of errors. Each job is an XLNT script containing the names of the programs to be executed as well as the input data. Also, after each program within a batch job, a special logging program is called which saves the program inputs and outputs as recorded in the console.

The BATCH program makes up the names of the input files based on the time, using even milliseconds, so that the names are unique. The log file has the same name as the job file with a suffix of LOG. The user is informed of the job name. BATCH attempts to find full network names for all files specified. If it can, the job is submitted with all file and directory names translated to the full network names so that it can run on any PC. Otherwise, it is submitted to a special queue which can only be run on the submitting PC, which then must have XLNT to run the job. The BATCH program also allows a special job to be run to stop a local queue after any currently running job is completed.

As an experiment, dual processor WINDOWS NT PCs have been placed in the Arkansas, Illinois, Mississippi, New Mexico, and North Dakota NASS state offices so that they can do their own large scale processing.

REFORMATTING SCENES

The PEDITOR system requires satellite scenes, no matter what their source, to be in a standard format for processing. The Landsat 5 and Landsat 7 scenes are received on CD-ROM in the fast format, somewhat different for the two satellites. Scenes from other sensors, when used, are received in various other formats. The reformatted data has the same pixel values and in the same coordinate system as the original data. The reformat converts the data to band interleaved by pixel format and adds a header. Also, registration information is derived from the header information supplied. This registration information may, or may not, be sufficient depending on the type of product ordered.

When submitting a reformat job, the user must first insert the CD-ROM containing the scene in a CD-ROM drive on the PC from which the reformat job is being submitted. If this CD-ROM has been made shareable, the reformat may be run on any PC set up to run batch jobs. Otherwise, it can only be run on the local PC. The user specifies the CD-ROM drive to be used, if more than one, the type of satellite data to be converted (the sensor type), and the output directory to contain the reformatted scene. The user may optionally exclude certain bands from the reformatted file, the default being to include all bands. BATCH then sets up the batch input file calling in the correct program for the type of scene specified. The output window file has a standard name which is supplied by BATCH and is used in further processing. This standard name is based on the scene identifier, which in turn is based on the path, row, and acquisition date. The reformat generally takes about 15 minutes and is mostly input/output intensive. An initial scene registration is also created based on information in the scene header. For a precision corrected scene, no other registration is required. Otherwise, a registration using the usual method of finding matching control points must be done interactively.

MULTITEMPORAL SCENE CREATION

A multitemporal scene is two scenes combined. Certain crops are easier to distinguish in a multitemporal scene than in a single scene. The two scenes must be over the same area and have the same pixel size, thus they must be from the same sensor, except that Landsat 5 and Landsat 7 scenes may be combined. One of the scenes is referred to as the primary scene and the other as the secondary scene. The coordinate system and registration of the multitemporal scene are those of the primary scene. The pixels from the secondary scene are matched to those from the primary scene. The nearest neighbor method is used. If the scene registration as obtained from the reformat step of the primary or secondary scene is not satisfactory, it is then necessary to register one or both before creating the multitemporal scene.

The methodology is to get a collection of 64 by 64 blocks from the primary scene on a regular grid. Then, the registrations of the two scenes is used to get corresponding locations for the block centers in the secondary scene. These block centers are then used to get centers for 32 by 32 blocks in the secondary scene. The blocks are read for a single band (currently band 2) and a gradient function is applied to all blocks. A correlation is performed for each possible position of the secondary block within the primary block with the selected position being that with the highest correlation. Once correlation is complete, block pairs are deleted for which the correlation is below a certain value or the shift is too great. Least squares polynomials are generated. The block pair with the highest residual is eliminated and the least squares polynomials are recomputed on the remaining blocks until the residual falls below a threshold or the number of block pairs is too small. If the residual is below the threshold and the number of block pairs is not too small, the mutltitemporal overlay is created by applying the least squares polynomials to each pixel in the primary scene to find the corresponding pixel in the secondary scene. One of the results of the correlation placed in the log file is a diagram showing which blocks remain in the correlation, allowing the user to check for suspicious results such as large areas where no block pairs remain. This procedure requires several programs to be

run, to obtain the block locations, read the blocks, do the correlation, and do the overlay.

The user specifies the directories containing the input window files and the scene identifier for the primary and secondary scene as well as the directory containing the output multitemporal window file. All files have standard names. The multitemporal job is then submitted with all other parameters having default values. It usually takes about 15 minutes with some portions being computationally intensive and others input/output intensive.

PACK AND CLUSTER

The PACK AND CLUSTER job creates statistics files which represent various covers and creates a final statistics file representing all covers of interest. These statistics files, which contain the means and variance covariance matrices are, with some possible modification, used for full scene classification.

First, segments must be associated with scenes. Each segment must be checked to verify that it is actually in the scene and also that it is useable in the scene, that is not cloud covered. Then each segment must be overlaid on the selected scene. This is a manual procedure. Then, each segment is converted into a raster format called a mask file so that there is an assignment between each pixel within the segment to a field or as a boundary pixel in two or more fields. Boundary pixels are ignored. For the pack and cluster and procedure to give useable results, it is, therefore, necessary that the overlay of the segment on the satellite data be accurate.

A packed file is a file containing only pixels from within selected fields in the segment. Two types of packed files are created, one containing pixels for all covers and then one for each cover with more than a fixed number of pixels and another for all covers combined with each having fewer than that number of pixels. This threshold value is initially set to a default value but may be changed by the user.

The packed files for the individual covers as well as those for the collection of covers with too few pixels to be created individually are then clustered using standard parameters. The result of clustering is a statistics file containing one or more categories for each cover. These parameters specify the starting and minimum number of clusters to output based on the number of pixels in the packed file as related to the total number of pixels. Each of these clusters is submitted as a separate batch job by calling the BATCH program in character mode, using a limited command set which is not available to the user in interactive mode. This is done so that clustering will run quicker if there are several PCs available since the size of the packed files varies widely and thus so does the time to do a cluster. The packed files have standard names based on the cover as do the resultant statistics files. A table is generated of all files to be clustered and this table is used to check if statistics files have been created for all of packed files. Each cluster job has in it the program executions for the next step, but it is only done if all statistics files are present. Thus the last cluster to complete does the next step.

Once all clusters are done, the job doing the last cluster makes a combined statistics file from all those created by the various clusters as well as an additional statistics file specified by the user, if any (see below). The packed file for all covers is then classified using the standard maximum likelihood algorithm. A tabulation is done on this classified file comparing the cover associated with the category assigned to the pixel as opposed to the reported cover for the pixel from the overlay of the segment on the satellite data. This is the percent correct tabulation. The user then reviews these results and decides if the statistics file generated is acceptable for full scene classification. If there are problems, the user may eliminate certain fields from consideration and try the PACK AND CLUSTER step again until satisfactory results are obtained. Some of the possible problems which can cause bad fields are the enumerator making a mistake in reporting a field or its boundaries or the segment being improperly overlaid on the satellite data

The user enters the scene names to be used for pack and cluster. Then, for each scene, the user enters the name of the window file containing the satellite data for all segments assigned to that scene. The BATCH program checks that the mask file is available for each segment and that the specified window file contains the segment. The user is then allowed to enter an additional statistics file. This statistics file, if used, typically contains categories which occur in the full scene but not in any segments. Some such categories are urban, water, and clouds. A category representing filler is added automatically. This is done so that when the classified scene is displayed, the area containing data is easily distinguishable from the filler. Filler area is, of course, not used in the estimation process so for estimation the assignment of filler is irrelevant. The statistics file created is then used for full scene classification, with possibly some editing by the user to eliminate questionable categories. The time required for a pack and cluster job can vary widely, depending on the time for the largest cluster job, which can be an hour or more

FULL SCENE CLASSIFICATION

Maximum likelihood classification is done on full satellite scenes, using all bands. Thus, for a multitemporal Landsat scene, 14 bands are used. The statistics file used for classification is that obtained from the PACK AND CLUSTER step above, with possibly some editing by the user. A maximum of 255 categories is permitted since each classified pixel is stored as one byte with the value of zero being set aside as an error indication. BATCH will not accept a statistics file with more than 255 categories. If such a statistics file is created, some categories must be deleted. Typically, the number of categories used is about 200 although this can vary.

Even with the fastest PCs available today, full scene classification is a lengthy procedure, often taking about 8 or 9 hours for a multitemporal scene, considerably less for a unitemporal scene. The classification program periodically saves the portion of the scene classified so far, allowing recovery in case of failure due to various hardware or software problems. In practice, this recovery feature is seldom used.

The user enters the name of the raw data window to be classified. It can be any window file and need not be a full scene, although a full scene is the most usual. The user is asked for the name of the statistics file. A check is made that the statistics file is for the same number of bands as the window file. The user is asked for the name of the output classified file with a suggested name being given. If the classified file exists, the assumption is made that this is a recovery from a failed classification.

CONCLUSIONS

Batch processing has allowed the faster PCs now available with WINDOWS NT to be used to run lengthy and repetitive jobs required for processing large amounts of satellite data covering entire or major portions of selected states with a minimum amount of user intervention. Due to various coordination issues, the assumption is made that the number of batch jobs is relatively low. If WINDOWS 2000 or one of its descendants includes a batch system, it will certainly be considered as a replacement. However, essentially the same method of creating batch jobs to do large scale processing will certainly continue into the foreseeable future in order to relieve the analyst from performing routine lengthy tasks over and over, leaving more time to concentrate on actual analysis of the satellite data.

REFERENCES

Allen, J.D.(1990) Remote Sensor Comparison for Crop Area Estimation Using Multitemporal Data. U.S. Department of Agriculture, NASS/R&AD SRB Staff Report Number SRB-90-03.

Bellow, M.E., Ozga, M. (1991) Evaluation of Clustering Techniques for Crop Area Estimation Using Remotely Sensed Data. In: *American Statistical Association 1991 Proceedings of the Section on Survey Research Methods*, Atlanta, Georgia, pp. 466-471.

Bellow, M.E., Graham, M.L. (1992) Improved Crop Area Estimation in the Mississippi Delta Region Using Landsat TM Data. In: *Proceedings of ASPRS/ACSM/RT'92 Convention*, Washington, D.C., pp. 423-432.

Graham, M.L. (1993) State Level Crop Area Estimation Using Satellite Data in a Regression Estimator. Survey Methods for Businesses, Farms, and Institutions, ICES Part I, U.S. Department of Agriculture, NASS Research Report No. SRB-93-10.

Hanuschak, G.A., Craig, M.E. (1993) Remote Sensing Program of the National Agricultural Statistics Service: From A Management Perspective. Survey Methods for Businesses, Farms, and Institutions, ICES Part II, U.S. Department of Agriculture, NASS Research Report Number SRB-93-11.

Ozga, M.. Processing Satellite Imagery on the INEL CRAY For Crop Area Estimation. Invited paper for the Sixth Annual INEL Computing Symposium, Evolutions in Technology, September 21-24, 1992, Idaho Falls, Idaho.

XLNT Reference Manual. Advanced Systems Concepts, Inc., Hoboken, NJ, USA.