Pattern Recognition

Speech, Image, Handwriting, etc.

Wajahat Qadeer
Rebecca Schultz
Ernesto Staroswiecki

Voice Recognition

- It is a process of automatically converting voice into its textual representation. This process encompasses two steps, which are namely:

Pre-Processing

This step consists of portioning and compression of speech into a stream of feature vectors. According to the prevailing benchmarks in this area, this part consumes 6.7% of the total recognition time. The widely used software for this part is RASTA which is also typically used as a benchmark for the evaluation of the pre-processing part.

Recognition

Once the stream of feature vectors is formed, these vectors are converted into a textual representation obtained by the identification of words that the vectors map to through the traversal of an optimal path of a graph. The state space that is searched is quadratic on the size of the vocabulary of the words. This step takes 93.3% of the running time of the total time of the recognition application. The widely used software for this part is SPHINX. This software is also used as a typical benchmark for the evaluation of the voice recognition soft wares.

VOICE RECOGNITION

Pre-Processing Step

- It is loop-oriented with fixed loop bounds.
- Every speech frame (typical size: 20ms of speech) is processed by this step and converted into feature vectors.
- Most of the inner loops present do not have any loop carried dependencies thus making them highly suitable for the exploitation of TLP.
- As a typical DSP application, it is computationally intensive requiring both floating point and integer operations.
- Because of the signal processing aspect there is a lot of DLP present in the typical applications used for the pre-processing of voice.
- This step has a small working set and memory foot print with regular access pattern.
- Within one frame of speech there is a high degree of spatial and temporal locality but it is limited to the particular frame being processed. If the frame size is increased, this property can be exploited otherwise a scratchpad instead of a cache is more suitable for the memory accesses.

VOICE RECOGNITION

- **Recognition Step**
  - It has a large working set with highly irregular control and data access patterns.
  - The main difficulty in the procedure is that the short sequences of frames may be interpreted locally in many different ways.
  - This step has a particularly big foot print especially in the initialization phase and requires very high memory bandwidth even in the steady state.
  - Large caches and bigger block sizes reduce cache misses by providing space for the exploitation of locality during the pruning process of the graph.
  - The amount of ILP present is not high, however, with some synchronization different search paths can be processed in parallel making this application suitable for TLP.
  - It has been shown that by exploiting different search algorithms, the data locality can be increased considerably.

VOICE RECOGNITION

	RASTA	SPHINX	gcc	gzip	vpr	mesa	art	equake	eon	twolf
Execution aggregates										
Instructions (10^9)	2.4	16.2	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Cycles (10^9)	4.0	25.4	0.7	0.4	0.6	0.7	1.0	0.7	0.7	0.7
IPC	0.6	0.6	0.7	1.3	0.8	0.8	0.5	0.8	0.8	0.8
Instruction mixes (%)										
Loads	26.6	23.9	14.8	22.7	18.2	10.2	21.5	20.1	18.3	9.8
Stores	6.9	6.4	7.3	7.1	8.6	6.1	9.6	4.0	25.9	9.1
Branches	8.8	14.3	13.4	10.8	15.4	14.3	4.4	12.0	14.3	14.0
Integer ops	42.2	50.5	63.1	59.3	47.5	47.7	39.9	34.8	25.6	59.3
FP ops	10.4	4.8	0.1	0.0	10.4	21.5	24.6	29.1	15.8	7.8
Branch misprediction rates = predictor hits / predictor updates (%)										
	5.3	9.4	9.6	11.3	7.2	3.6	10.5	3.4	9.7	4.1
Cache and TLB miss rates (%)										
DL1	0.5	15.8	2.5	7.9	2.2	0.6	32.7	0.1	0.1	0.6
IL1	3.4	3.2	14.7	10.1	17.3	19.3	0.1	15.4	16.1	16.7
L2	1.9	41.9	4.4	32.9	3.1	0.8	71.7	0.5	0.1	1.4
DTLB	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure1: Comparison with SPEC2000 benchmarks

VOICE RECOGNITION

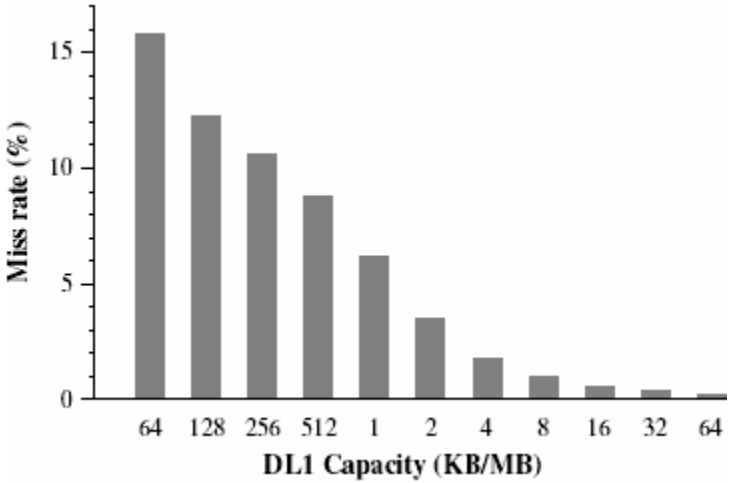


Figure2: Miss rate vs. data L1 cache

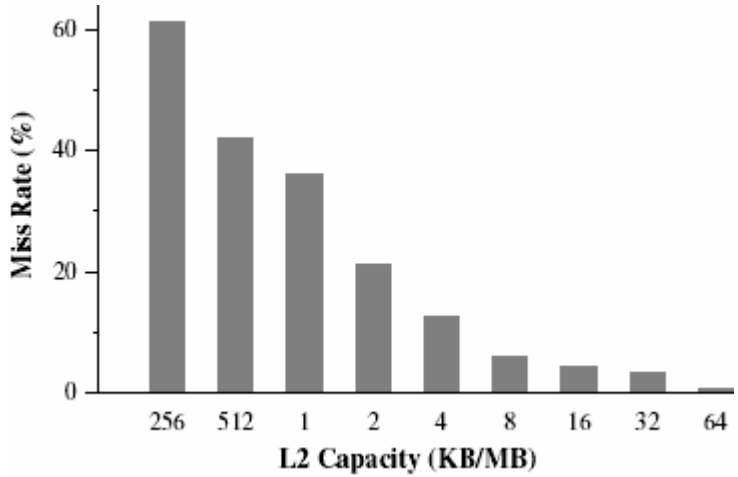


Figure2: L2 Miss rate vs. capacity with a constant 64KB data L1 cache

VOICE RECOGNITION

- **Algorithms for Voice Recognition:**
  - Although, there are many algorithms which can be used for voice recognition namely, Dynamic Time Wrapping, Neural Networks, Hidden Markov Modeling etc, the one that is widely being used due to its better results is Hidden Markov Modeling. Most of the current research is based on the modification of Markov Modeling based algorithms.
- **Scaling Trends**
  - New applications in speech recognition are concentrating on increasing the vocabulary of the words. The complexity of the current search algorithms, which are based on the hidden Markov Models, increases quadratically with the size of the vocabulary.

VOICE RECOGNITION

- **References**

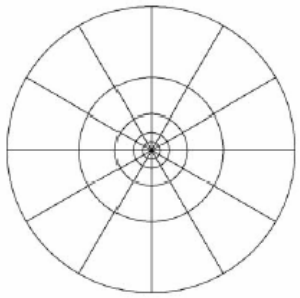
  - [1] A Characterization of Speech Recognition on Modern Computer Systems, *K. Agaram, S. W. Keckler and Doug Burger, Proceedings of 4th annual workshop on Workload Characterization, 2001.*
  - [2] Performance Analysis of Speech Recognition Software, *C. Lai, S. Luand Q. Zhao, Workshop on Computer Architecture Evaluation using Commercial Workloads, 2002*

IMAGE RECOGNITION

- Also a 3-step process:
  - **Edge detection:**
    Filtering
  - **Image processing / Characterization**
  - **Matching**

IMAGE RECOGNITION

- Processing / Characterization



  - We need to find image descriptors:
    Shape contexts, Fourier descriptors, etc.
  - Similar characteristics to voice recognition preprocessing except:
  - Not necessary to use floating point or excessive computation,
  - Yet more points to look at, which grow with the size of the image,
  - And although the memory access pattern is very regular, is important to remember that now we are looking at a 2D window.

IMAGE RECOGNITION

- Matching
  - Once again, similar to voice recognition, but problems really exacerbated!
  - Several algorithms: SVMs, Shortest Augmenting Path, etc
  - Remember that dictionary must be MUCH larger
  - Little ILP, some DLP, but mostly TLP
  - Topics to explore: CAMs, prefetching (but be careful!)

IMAGE RECOGNITION

- Additions:
  - Scalability:
    - The matching algorithms are similar to those of voice recognition, and at least some of the graph oriented algorithms are $O(V^2)$, where V is the number of vertices in the graph. Unlike voice recognition, where the number of vertices depends only on the size of the dictionary, here it depends on both the size of our learning set (dictionary) AND the size of the image.
    - When we increase the size of the image, we need more reference points to use as descriptors, since we can now fit more objects in an image, and we might need to decide with finer resolution. Note that this increases both the number of points we need to match and the number of points we need to store.
    - The dictionary for object recognition can be virtually infinite. Unlike voice or handwriting recognition, where there is a finite number of useful phonemes or symbols, there is no such limitation for images. This means that we can be using orders of magnitude larger memory, and therefore the number of memory accesses could be too large, taking most of the runtime.

IMAGE RECOGNITION

- References:
  - UC Berekeley Computer Vision Group:
    http://http.cs.berkeley.edu/projects/vision/vision_group.html
    (In particular: **Shape Matching and Object Recognition**)
  - LIBSVM FAQ:
    http://www.csie.ntu.edu.tw/~cjlin/libsvm/faq.html
- One observation:
  - The "Processing" step of image recognition uses techniques from standard multimedia and signal processing applications.
  - The "Matching" step uses techniques from standard Machine Learning applications.
  - So look at their presentations too!

HANDWRITING RECOGNITION

- Special case of image recognition
  - Some on-line algorithms may also use some data about the strokes of the letters
- Similar algorithms for selecting descriptors and matching
  - Neural Nets, Hidden Markov Models, etc
- Matching library is small and fixed size
  - Consists of a few ways of writing each character
- Rarely done in hardware
  - On-line solutions need run no faster than you can write
- Scaling
  - Constant number of descriptor points irrespective of sample size
    - Regardless of the size of the characters or the total amount of text the appropriate number of descriptors is constant. This is not true for general image recognition
- Limited opportunities for extensions

HANDWRITING RECOGNITION REFERENCES

- Young-Joon Kim, Seong-Whan Lee, Myung-Won Kim, "Parallel hardware implementation of handwritten character recognition system on wavefront array processor architecture" *Third International Conference on Document Analysis and Recognition (Volume 2),* Montreal, Canada, August 1995.
- J. Hu, S. G. Lim and M. K. Brown, "Writer independent on-line handwriting recognition using an HMM approach" *Pattern Recognition*, January 2000.
- Jianying Hu, Sok Gek Lim and Michael K. Brown, "HMM based writer independent on-line handwritten character and word recognition", *The 6th International Workshop on Frontiers in Handwriting Recognition*, KAIST Campus, Taejon city, Korea, August 1998.