

Finding a Ph.D. Research Topic

via Literature Search and by Reading Scientific Papers

Olli Silvén
University of Oulu



Machine Vision Group
Dept. of Electrical and Information Engineering, and Infotech Oulu

Introduction (1)

Fast forward:

When your Ph.D. thesis manuscript is submitted to the reviewers, they will determine, whether

- the research has been worth doing,
- you have created new scientific results, and
- the presentation is clear and adheres to scientific conventions

If all these findings are positive, they will end their statements “...as the conclusion I recommend granting Ms./Mr. X the permission to publish the manuscript and to defend it publicly as her/his doctoral thesis”

Fast backward:

Finding an appropriate research topic is the most important problem, and often the biggest obstacle for a beginning graduate student...

Introduction (2)

If selecting the research topic is so important, why to leave that into the hands of an inexperienced graduate student? Indeed...

If your professor/thesis advisor is active in his field

- he/she should have plenty of research ideas in his area,
- should be able to point out a rough topic to start with,
- should be able to sketch the expected scientific findings,
- and have a back-up plan if the initial topic runs into difficulties

Good to know:

- over 80% of doctoral theses are made in research groups
- research groups are likely to provide stable funding
- but life with a lone-wolf professor can be exciting
- (do I really need the professor?)

Introduction (3)

The characteristics of an ideal research problem are incompatible

Compactness:

- previous research should have brought your problem close to a kill
- = new scientific results quickly!
- ...but someone can be even quicker!

Significance:

- aim high, e.g., cross-breed two or more fields of science
- ...but risk finishing late or never
- and having few people with whom to discuss your research



Machine Vision Group

Dept. of Electrical and Information Engineering, and Infotech Oulu

Introduction (4)

Chances for creativity:

- the best Ph.D. theses have a high degree of creativity
- ...but can you count on yourself in that area?

Ready-made multi-year research plan:

- nothing is easier than following a roadmap
- ...but it often unclear at first how the ideas will develop

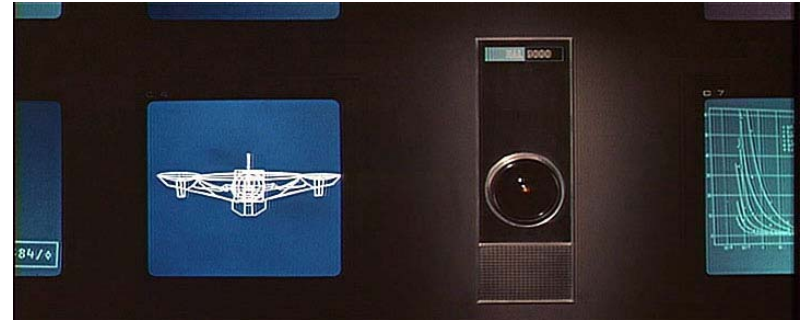
BTW: characterize an ideal Ph.D. thesis!

Looking for PhD thesis topic ideas from science fiction: HAL 9000 technology on "Discovery" spaceship

(in Arthur C. Clarke's 2001: A Space Odyssey; 1968)

Capable of

- face detection and recognition
- lip reading
- Interpreting emotions
- speech recognition & natural language processing
- learning and logical reasoning



HAL 9000 was proposed 1965 and supposed to be technology available in 1997...

- when did you expect your PhD thesis get finished?

Transporter technology on the "Enterprise" spaceship

(from Star Trek)

Capable of transferring living objects

- particle beams and quantum technology?

This technology is expected to become available during the 24th century – if ever

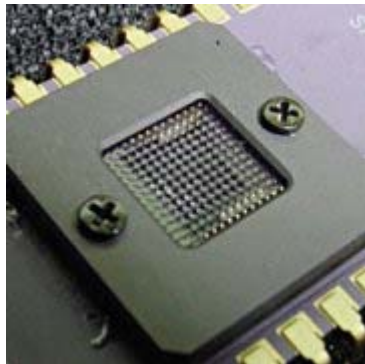
When was the deadline for your PhD thesis?



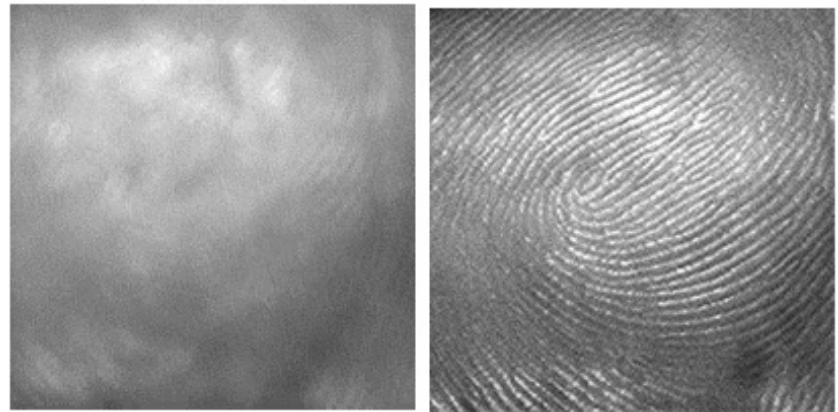
Perhaps more suitable goals for PhD thesis research

- :
- computing at almost no cost and energy!
- unlimited communications bandwidth at almost no energy
- printable electronics and printable cameras

Notice: no tangible goal,
but somehow measurable/comparable steps toward the grand goal



Lenslet camera (Tamida,J.
University of Osaka)

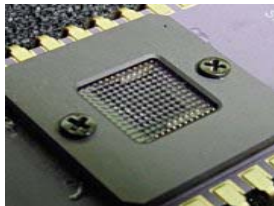
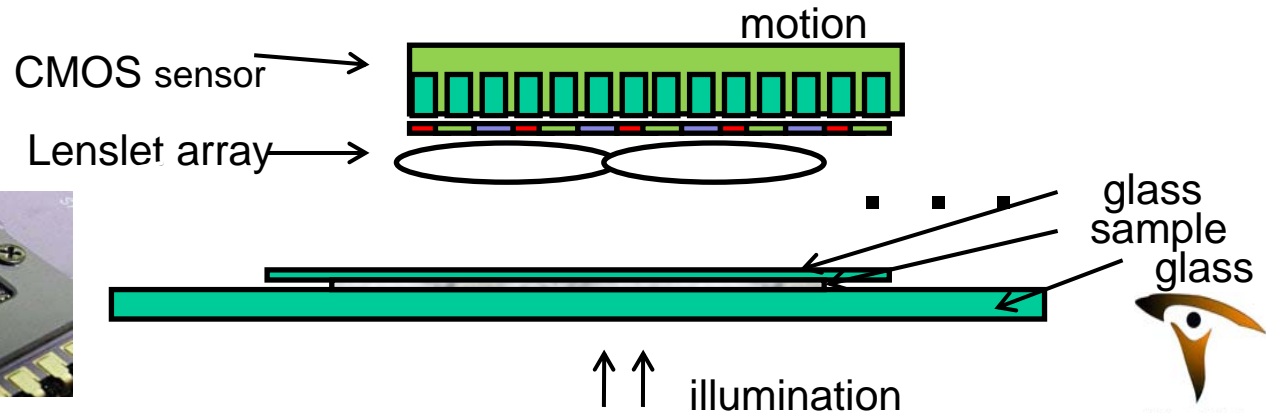
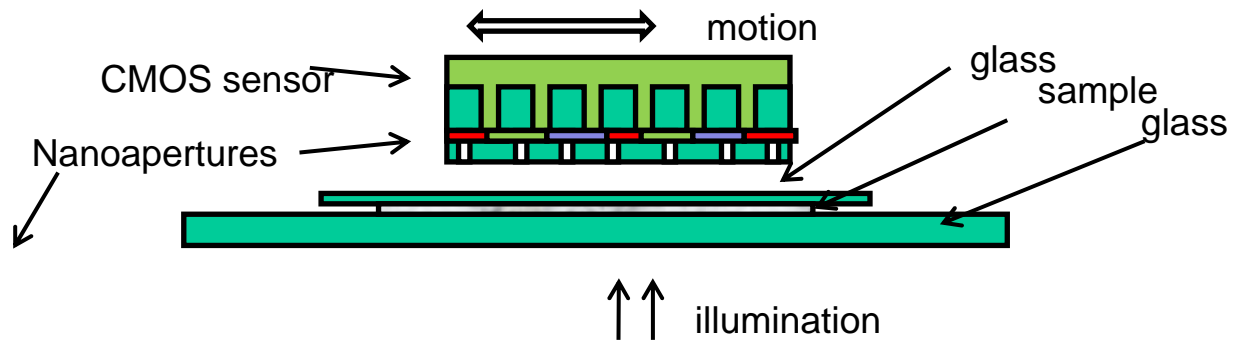
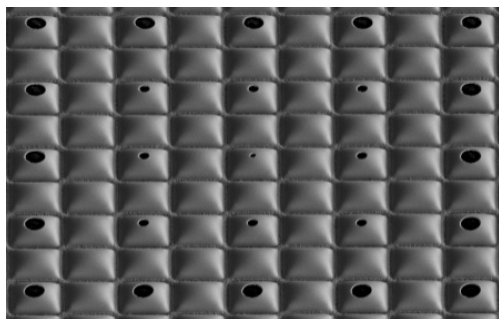


Imaging with wavefront coding
(CDM Optics)

Derivative idea: "Flat imaging" technologies for microscopy

In development

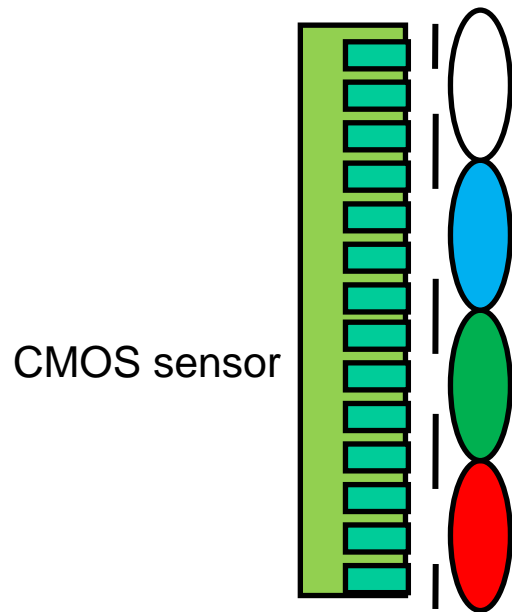
- nano-apertures or lenslet array on Bayer patterned sensor
- backend computing in a mobile device
- combined super-resolution, panorama and high dynamic range imaging



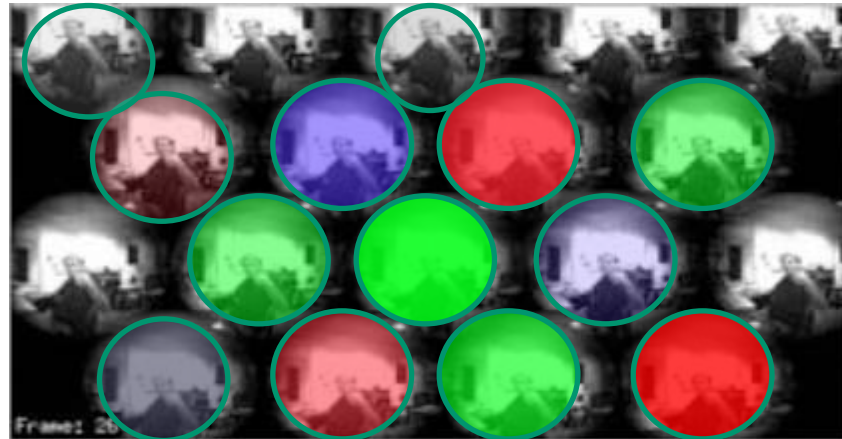
Derivative idea: "Flat imaging" for ubiquitous computing

In development

- Lenslet array with colour filters & different apertures
- face detection and recognition as initial backends
- the ultimate ubiquitous camera solution



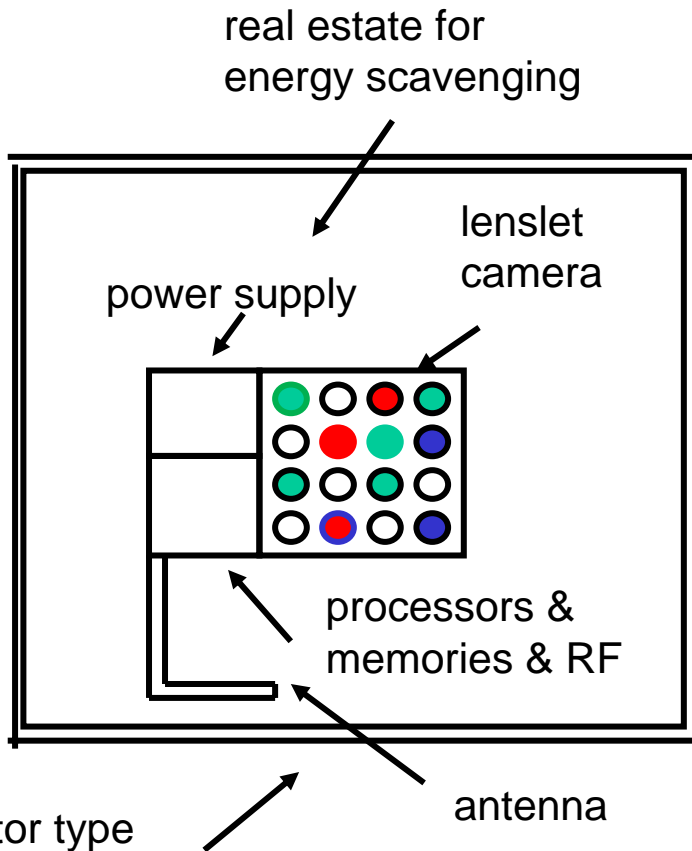
Lenslet array



The challenges?

Towards HAL 9000i

Intelligent, zero-power, flat, sticker like camera unit



Capable of

- face detection and recognition
- lip reading
- Interpreting emotions
- speech recognition

All you need to know in the beginning (1)

Almost nothing is new

- first, figure out your idea or what you have done,
- then go to the electronic data bases and find it there,
- and tell your findings to your professor, in your papers, and Ph.D. thesis

Write your reports, papers and thesis in the order of decreasing believability to ensure the maximum reusability for yourself

- first, describe the research problem
- then tell what the earlier peer-reviewed results are
- add your own results and methodology
- only "discussion" can contain unverified hypotheses

Learn the scientific use of language

- "obviously, probably, certainly, undoubtedly"
= you don't know what you are talking about

All you need to know in the beginning (2)

Never jump into the conclusions

- always write "discussion" before conclusions

Don't write detective stories

- always tell your novel results in the beginning of your presentations
- ... and use the remaining time to prove their validity

Mine all the gold and ask the right questions

- try to extract everything you can from your data
- ...and try to see whether someone has not done that for his data

Most important: generate ideas and work on them

- be creative
- ...don't waste **all** your time reading other people's papers
- but don't fall in love with your first own idea

"Chance favors a prepared mind" (Louis Pasteur)

Application of the theory: examples on finding research topics

Example 1: Transform domain motion estimation for video coding

- new ideas on old foundations
- public defence 2007

Example 2: Dynamic strength grading of wood from image data

- cross-breeding of material and computer science
- public defence ~2011

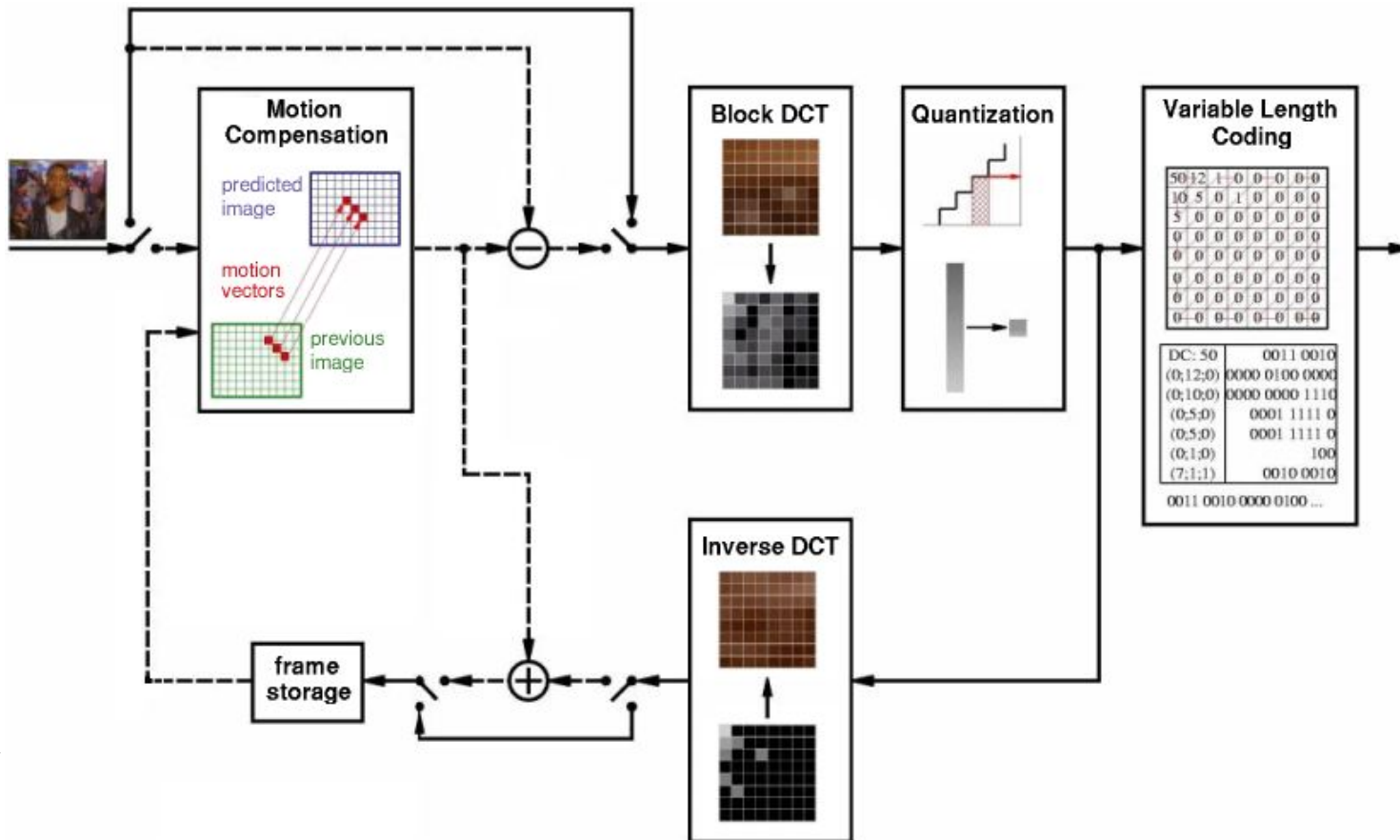
Example 3: Energy-efficient architectures for mobile devices

(under this title to avoid suspicions of megalomania...)

- trying to change the paradigms of computer and software architectures
- public defences 2009, ~2010, etc.

Example 1: transform domain motion estimation -background

Most video coding standards rely on hybrid block based transform coding methods
A key technique in achieving good compression efficiency: motion estimation

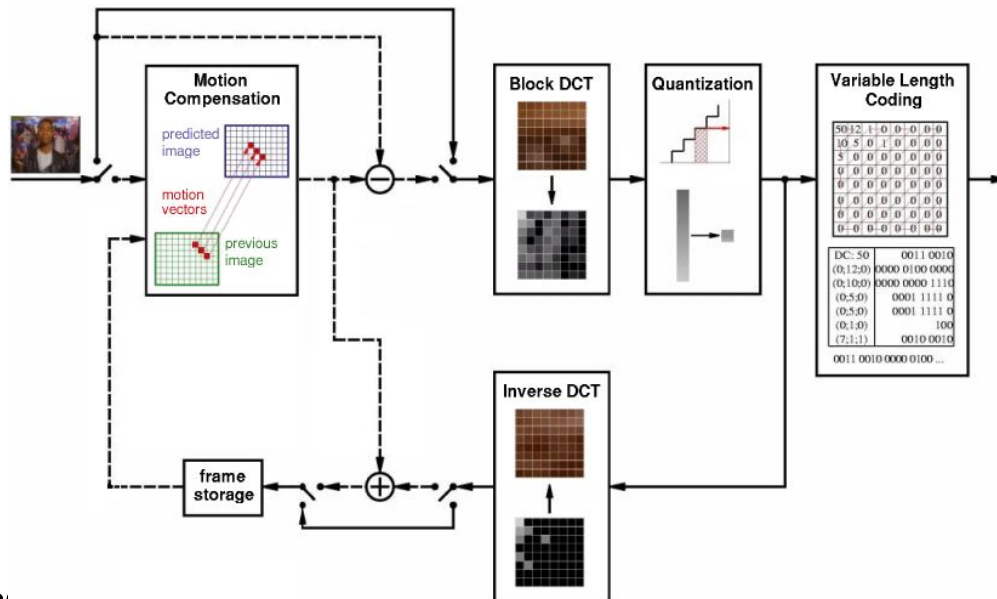


Example 1: Motion estimation for video coding

The best motion estimation method is "full-search" via correlation, but it is very expensive [$O(M \cdot M \cdot N \cdot N)$]

Other techniques compromise compression efficiency, but reduce computational complexity by being partial search solutions

Hypothesis: low cost implementation of full-search is possible



Example 1: Initial literature search:

Let's see, what is the state of the art... (around 2003)

IEEExplore: motion estimation <and> video coding

→ over 1000 references!

IEEExplore: motion estimation <and> video coding <and> full search

→ less than 200 references

First scan-through and preliminary classification of full search approaches:

Spatial domain search techniques

Exhaustive search: computational cost overwhelming

Partial distortion elimination: non-deterministic time

Successive elimination: non-deterministic time

(+numerous non-full-search variants (3SS etc.))

Transform domain search techniques

DCT domain search: deterministic time, complexity?

Fast correlation via FFT/IFFT: deterministic time

Example 1: Is this field really fully understood?

Key observations

Transform domain techniques appear to be less known

- possible reasons:

 - fast correlation via FFT/IFFT requires complex arithmetics

 - >18 bits of dynamic range needed when images are 8-bits/pixel

The Fourier transform is not the only transform for which the convolution/correlation theorem can be applied!

→ refined hypothesis:

"low" computational cost implementation of full search motion estimation is possible via transform domain techniques

Example 1: Key facts in brief

Common knowledge:

Cooley-Tukey FFT: $O(N \log(N))$

-> Convolution/correlation: $O(N \log(N))$

Less common knowledge:

Winograd: Convolution/correlation Lower Bound $O(2N-1)!$

→ Winograd Convolution/Correlation: $O(2N-1)$

→ Number Theoretic Convolution/Correlation: $O(2N-1)$

(How do we know this? It is in the *older* textbooks, stupid!)

Example 1: Has someone already done this?

IEEE Explore: motion estimation <and> video coding <and> full search
<and> transform

→ 13 references, apparently no exact match

[the nearest one: Koc & Liu (1998) DCT based motion estimation]

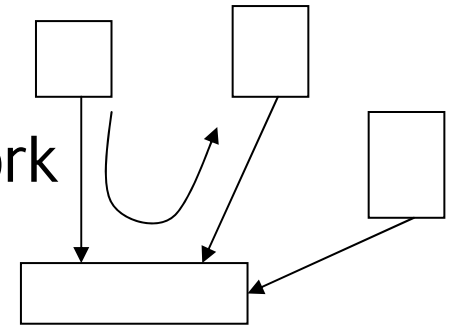
Now to the research plan! **We appear have an idea of our own!**

Find whether it makes sense, investigate the
key issues and publish/patent! Quickly, anyway...

The research plan is needed to

- build a roadmap for ourselves
- and to enable using master's thesis students for selected parts

Example 1: Research plan (1): start of the the hard work



1. Learn the relevant background:

R. Blahut, *Fast Algorithms for Digital Signal Processing*

H. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*

S. Burrus and T.W. Parks, *DFT/FFT and Convolution Algorithms*

... and how it has been used

= find relevant references from citation databases and track the citations, and build your own reference data base; write your own survey

2. Thoroughly analyse other full-search approaches and the best non-full-search approaches; re-implement when necessary.

→ you will learn, what was not explicitly told in the papers

Example 1: Literature search

Checking the references to the textbooks of the field using the Web of Science

Blahut R, Fast algorithms..., 213 references...

relevant ones: Bernardini et al 1996; Ran & Liu 1995, Stone & Wolpov 2002,
but none of these directly in our domain of interest

Nussbaumer HJ, Fast Fourier..., 172 references, only one relevant one that wasn't
found previously: Stasinski & Nandi 1992

.....

It is clear that we now need to concentrate on the relevant references,
starting with Koc & Liu 1998 that we will read with the refereeing mode on

Example 1: Research plan (2)

3. Analyse variations of the fast correlation approaches; implement, experiment, and document!

Decide evaluation criteria: word length, operation count, chip area, memory BW...

Hopefully, get a great idea, otherwise add understanding via meticulous analysis of the methods and experiments

4. Write papers (and your Ph.D. thesis)

Example 1: Literature search (cont'd)

From the short list of IEEExplore: Koc & Liu 1988: DCT based motion estimation

- computational complexity $O(M^2)$ for $M \times M$ search range!

DCT-Based Motion Estimation

Ut-Va Koc, *Member, IEEE*, and K. J. Ray Liu, *Senior Member, IEEE*

Abstract—We propose novel discrete cosine transform (DCT) pseudophase techniques to estimate shift/delay between two one-dimensional (1-D) signals directly from their DCT coefficients by computing the pseudophase shift hidden in DCT and then employing the sinusoidal orthogonal principles, applicable to signal delay estimation remote sensing. Under the two-dimensional (2-D) translational motion model, we further extend the pseudophase techniques to the DCT-based motion estimation (DXT-ME) algorithm for 2-D signals/images. The DXT-ME algorithm has certain advantages over the commonly used full search block-matching approach (BKM-ME) for application to video coding despite certain limitations. In addition to its robustness in a noisy environment and low computational complexity, $O(M^2)$ for an $M \times M$ search range in comparison to the $O(N^2 \cdot M^2)$ complexity of BKM-ME for an $N \times N$ block, its ability to estimate motion completely in DCT domain makes possible the fully DCT-based motion-compensated video coder structure, which has only one major component in the feedback loop instead of three as in the conventional hybrid video coder design, and thus results in a higher system throughput. Furthermore, combination of the DCT and motion estimation units can provide space for further optimization of the overall coder. In addition, the DXT-ME algorithm has solely highly parallel local operations and this property makes feasible parallel implementation suitable for very large scale integration (VLSI) design. Simulation on a number of video sequences is presented with comparison to BKM-ME and other fast block search algorithms for video coding applications even though DXT-ME is completely different from any block search algorithms.

Index Terms—Discrete cosine transform, motion estimation, shift measurement, time delay estimation, video coding.

dimensional (1-D) signals, a common problem in many areas of signal processing such as time delay estimation [7], [8] and optical displacement measurement [9]. As a matter of fact, shift estimation for 1-D signals and translational motion estimation for 2-D images inherently address the same problem and can use similar techniques to approach.

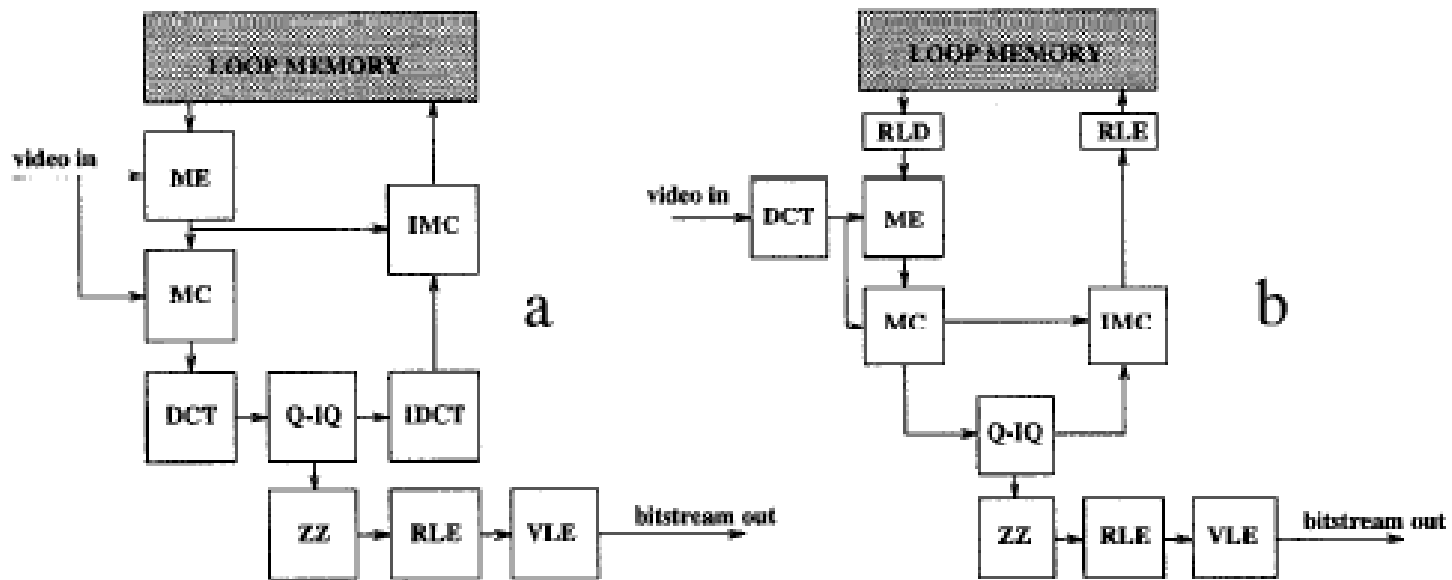
In video coding, the most commonly used motion estimation scheme is the full search block-matching algorithm (BKM-ME), which searches for the best candidate block among all the blocks in a search area of larger size in terms of either the mean-square error [10] or the mean of the absolute frame difference [11]. The computational complexity of this approach is very high, i.e., $O(N^2 \cdot M^2)$ for an $N \times N$ block in an $M \times M$ search range. Even so, BKM-ME has been successfully implemented on VLSI chips [3]–[5]. To reduce the number of computations, a number of suboptimal fast block-matching algorithms have been proposed [10]–[15]. However, these algorithms require three or more sequential steps to find suboptimal estimates. Recently, a correlation-based approach [16] using complex lapped transform (CLT-ME) to avoid the blocking effect was proposed, but it still requires searching over a larger search area and thus results in a very high computational burden. Moreover, motion estimation using the CLT-ME is accurate on moving sharp edges but not on blur edges.

In addition to block-based approaches, pel-based estimation methods such as pel-recursive algorithm (PRA-ME) [17], [18]

Example 1: Literature search (cont'd)

Check from Web of Science (ISI citation database) which papers in-turn refer to Koc & Liu 1998: DCT based motion estimation:

- one relevant reference: Kleihorst & van der Vleuten (2000): DCT-domain embedded memory compression for hybrid video encoders (this in turn was not (in 2003) referenced by recent relevant papers)



Example 1: Analysis

Understand the purpose of the paper

- = read the abstract and introduction;
the claimed contributions are stated there

Abstract—We propose novel discrete cosine transform (DCT) pseudophase techniques to estimate shift/delay between two one-dimensional (1-D) signals directly from their DCT coefficients by computing the pseudophase shift hidden in DCT and then employing the sinusoidal orthogonal principles, applicable to signal delay estimation remote sensing. Under the two-dimensional (2-D)

Example 1: Analysis (cont'd)- the idea = the key contribution

From the introduction:

In this paper, we present new techniques called the *DCT pseudophase techniques* [26], [27] applicable to delay estimation for 1-D signals or motion estimation for 2-D images. Unlike other fast block search motion estimation methods (such as logarithmic, three-step search, cross, subsampled

Example 1: Analysis (cont'd) - the value of contributions

methods. In addition to low computational complexity (on the order of M^2 compared to $N^2 \cdot M^2$ for BKM-ME for the search range M and block size N) and robustness of the DCT pseudophase techniques, this algorithm takes DCT coefficients of images as input to estimate motion. Therefore, it can be incorporated efficiently with the DCT-based coders used for most current video compression standards as the fully DCT-based video coder structure. It enables combining both the DCT and motion estimation into a single component to further reduce the coder complexity and at the same time increases the system throughput as explained in details in Section IV. Finally, due to the fact that the computation of pseudo phases involves only highly local operations, a highly parallel pipelined architecture for this algorithm is possible.

Example 1: Analysis (cont'd) - the method

$$\begin{bmatrix} X_2^C(k) \\ X_2^S(k) \end{bmatrix} = \begin{bmatrix} Z_1^C(k) & -Z_1^S(k) \\ Z_1^S(k) & -Z_1^C(k) \end{bmatrix} \begin{bmatrix} g_m^C(k) \\ g_m^S(k) \end{bmatrix} \quad (3)$$

where $g_m^C(k) \triangleq \sin[(k\pi/N)(m - (1/2))]$ and $g_m^S(k) \triangleq \cos[(k\pi/N)(m - (1/2))]$ are called *pseudophases*

Example 1: Analysis (cont'd)

- the conclusions based on experiments

In this paper, we present new DCT pseudophase techniques utilizing the concept of pseudophase shifts hidden in the DCT coefficients of shifted signals and the sinusoidal orthogonal principles for estimation of shift/delay completely in the DCT domain. In extension to the 2-D case, the DCT pseudophase techniques result in the DXT-ME algorithm, a DCT-based motion estimation algorithm. We show that this DXT-ME algorithm exhibits good estimates even in a noisy situation. Due to its capability of motion estimation in the DCT domain, its application to video coding realizes the fully DCT-based motion-compensated video coder structure which contains, in the performance-critical feedback loop of the coder, only one major component, the transform-domain motion estimation unit instead of three major components as in the conventional

Example 1: Read the paper in refereeing mode (1)

Koc & Liu 1998: DCT based motion estimation

Walkthrough:

1. Purpose of the paper, is it clear? First page, last paragraph

In this paper, we present new techniques called the *DCT pseudophase techniques* [26], [27] applicable to delay estimation for 1-D signals or motion estimation for 2-D images.

2. Appropriateness for the intended forum? Yes
3. Has the work been worth doing? $O(N^2M^2) \rightarrow O(M^2)$!
4. Is the method of approach clear and valid? Chapter II, Equation (3)

$$\begin{bmatrix} X_2^C(k) \\ X_2^S(k) \end{bmatrix} = \begin{bmatrix} Z_1^C(k) & -Z_1^S(k) \\ Z_1^S(k) & +Z_1^C(k) \end{bmatrix} \begin{bmatrix} g_m^C(k) \\ g_m^S(k) \end{bmatrix}$$

Example 1: Read the paper in refereeing mode (2)

Koc & Liu 1998: DCT based motion estimation

Walkthrough:

5. Is the actual execution of the research correct? Check the math and learn
6. Are the conclusions correct? In principle, yes. Real applicability?

completely in DCT domain makes possible the fully DCT-based motion-compensated video coder structure, which has only one major component in the feedback loop instead of three as in the conventional hybrid video coder design, and thus results in a higher system throughput. Furthermore, combination of

7. Is the presentation satisfactory? Yes, could be more compressed
8. What did you learn? **Look carefully at the references!**

Example 1: Analyse the references (below start of a list of 41)

- [1] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images—A review," *Proc. IEEE*, vol. 76, pp. 917–935, Aug. 1988.
- [2] H. G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, pp. 523–548, Apr. 1993.
- [3] K. M. Yang, M. T. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1317–1325, Oct. 1989.
- [4] T. Komarek and P. Pirch, "Array architectures for block-matching algorithms," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1301–1308, Oct. 1989.
- [5] L. D. Vos and M. Stegherr, "Parametrizable VLSI architectures for the full-search block-matching algorithm," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1309–1316, Oct. 1989.
- [6] R. C. Kim and S. U. Lee, "A VLSI architecture for a pel recursive motion estimation algorithm," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1291–1300, Oct. 1989.
- [7] *IEEE Trans. Signal Processing, Spec. Issue Time Delay Estim.*, vol. ASSP-29, 1981.
- [8] G. Jacovitti and G. Scarano, "Discrete-time techniques for time-delay estimation," *IEEE Trans. Signal Processing*, vol. 41, pp. 525–533, 1993.
- [9] J. P. Fillard, J. M. Lussert, M. Castagne, and H. M'rimet, "Fourier phase

Example 1: Analysis of the references

Koc & Liu 1998: DCT based motion estimation

- 1 Check what is found using the key words of the article: not much...
- 2 Study the key references you don't know beforehand:
Transform domain methods: Kuglin & Hines 1975, Thomas 1987, Girod 1993
Young & Kingsbury 1993, (this reader would add Lees & Henshaw 1986 Proc. SPIE 730)
- 3 Check which recent papers (until 1998) cite the same references
No similar ideas found, just the ordinary "whatever you do, I can do better"
- 4 Check the references of those recent papers

Example 1: Analysis of the references (cont'd)

Who has cited Kuglin & Hines 1975? (... can't find this, ISI database only from 1986...)

Who has cited Thomas 1987 (... can't find this, it is a conference paper)

Who has cited Girod 1993? (seven references till 1997) Examples

Girod B, Steinbach E, Farber N [Performance of the H.263 video compression standard](#) J VLSI SIG
PROCESS S 17 (2-3): 101-111 NOV 1997

Ku CW, Chen LG, Chen CH, et al. [Investigation of a visual telephone prototyping on personal computers](#)
IEEE T CONSUM ELECTR 42 (3): 750-759 AUG 1996

Kuo CM, Hsieh CH, Jou YD, et al. [Motion estimation for video compression using Kalman filtering](#)
IEEE T BROADCAST 42 (2): 110-116 JUN 1996

KOKARAM AC, MORRIS RD, FITZGERALD WJ, et al. [INTERPOLATION OF MISSING DATA IN IMAGE SEQUENCES](#) IEEE T IMAGE PROCESS 4 (11): 1509-1519 NOV 1995

.....

Conclusion: our idea appears to be a novel one-> we can contribute!

Example 1: A paper published on an other forum 1996 (not cited by Koc & Liu 1998)

Motion Estimation in DCT Domain[†]

Mingzhou Song, Anni Cai, *Member, IEEE* and Jing-ao Sun, *Member, IEEE*
Radio Engineering Department, Beijing University of Posts and Telecommunications

Internal P. O. Box 57
Beijing Univ. of Posts and Telecommunications
Beijing, 100088, P. R. China

E-mail: jingsun@bupt.edu.cn

Abstract — In this paper, a new block matching criterion is proposed for motion estimation in video coding. The new criterion is based on a comparison between the discrete cosine transform (DCT) coefficients of two blocks to be matched. Since the spectrum of DCT statistically concentrates on the neighborhood of the *dc* component and the number of non-zero coefficients is quite small, only few coefficients need to be considered when matching blocks according to the new criterion. Almost in all the video coding standards, DCT is a necessary step for spatial redundancy reduction. Thus the generated DCT coefficients can be utilized in the new criterion. However, there are still some blocks whose DCT coefficients are not available. Several algorithms of calculating DCT coefficients of these blocks are given. When this new criterion is combined with the logarithm search, promising results are produced.

coefficients can be utilized into motion estimation, so the re-calculation of DCT coefficients can then be made ready for use in the proposed matching criterion. However, there do not exist ready-made DCT coefficients of all the candidate blocks in the reference frame. Only those blocks both horizontally and vertically located at integral multiple of the block size have ready-made DCT coefficients while the others do not. We call the former aligned blocks, the latter non-aligned blocks. For the non-aligned blocks, the DCT coefficients can be derived directly by DCT from samples in spatial domain or indirectly by the known DCT coefficients of the adjacent aligned blocks. Kou et al. and Chang et al. discussed the indirect approaches [3],[4], which will be reviewed in section III. Nevertheless, non-aligned blocks need extra computations which may scratch off the efficiency gained by using the new criterion. So searching

Example 1: Full list of references

- [1] Draft International Standard MPEG-I, ISO/IEC JTC 1/SC 29-N 071, CD11172, Dec. 1991.
- [2] Draft International Standard MPEG-II, ISO/IEC DIS 13818, 1994.
- [3] Shih-Fu Chang and David G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video", IEEE J. Select. Areas Commun., Vol. 13, no. 1, pp. 1-11, Jan. 1995.
- [4] Weidong Kou and Tore Fjällbrant, "A direct computation of DCT coefficients for a single block take from two adjacent blocks", IEEE Trans. on Signal Processing, Vol. 39, no. 7, pp. 1692 - 1695, July 1991.
- [5] H. G. Musmann, P. Pirsch and H-J. Grallert, "Advances in picture coding", Proceedings of the IEEE, Vol. 73, no. 4, pp. 523 - 548, April 1985.
- [6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, "Motion-compensated interframe coding for video conferencing", in NTC 81 proc., pp. G5.3.1 - G5.3.5, 1981.
- [7] W.-H. Chen, C. H. Smith and S. C. Fralick, "A fast algorithm for the discrete cosine transform", IEEE Trans. on Commun., vol. COM-25, pp. 1004 - 1009, Sept. 1977.

Your quick verdict??? You have to read the papers to judge!

Example 1: Partial list of publications and reports based on the research topic

- Ojansivu V, Silvén O, Huotari R & Toivonen T (2003) [Digital video quality evaluation method](#). Proc. of the Finnish Signal Processing Symposium (FINSIG 2003), May 19, Tampere, Finland, 115-119.
- Toivonen T (2002) [Number theoretic transform-based block motion estimation](#). M.Sc. thesis, Department of Electrical Engineering, University of Oulu, Finland.
- Toivonen T & Heikkilä J (2003) [A new rate-minimizing matching criterion and a fast algorithm for block motion estimation](#). Proc. IEEE International Conference on Image Processing, September 14-17, Barcelona, Spain, 2:355-358.
- Toivonen T & Heikkilä J (2003) [Efficient method for half-pixel block motion estimation using block differentials](#). In: Visual Content Processing and Representation, VLBV 2003 Proceedings, Lecture Notes in Computer Science 2849, Springer, 225-232.
- Toivonen T & Heikkilä J (2004) [Fast full search block motion estimation for H.264/AVC with multilevel successive elimination algorithm](#). Proc. IEEE International Conference on Image Processing (ICIP 2004), Singapore, 1485-1488.
- Toivonen T & Heikkilä J (2006) [Video filtering with Fermat number theoretic transforms using residue number system](#). IEEE Transactions on Circuits and Systems for Video Technology 16(1): 92-101.
- Toivonen T & Heikkilä J (2005) [Reduced frame quantization in video coding](#). Proc. International Workshop on Very Low Bit-rate Video (VLBV05), Costa Rei, Sardinia, Italy, 4 p.
- Toivonen T, Heikkilä J & Silvén O (2002) [A new algorithm for fast full search block motion estimation based on number theoretic transforms](#). Proc. 9th International Workshop on Systems, Signals and Image Processing, November 7-8, Manchester, United Kingdom, 90-94.
- Toivonen T, Heikkilä J & Silvén O (2003) [Menetelmä ja laite peräkkäisten kuvien koodaamiseksi \(Method and device for coding successive images\)](#). Suomalainen patentti no. 111592 (A Finnish patent).

Example 2: Dynamic strength grading of lumber - background

Strength graded lumber is used for house frames, roof structures etc.

Conventional strength grading is based on measuring the modulus of elasticity

- Bending tests can be destructive, strength correlation $r^2 \sim 0.4-0.5$

Lots of research work on strength grading has been carried out using at least

- microwaves
- x-rays
- computer vision
- ultrasound
- bending

Still, at most $r^2 \sim 0.6$ has been achieved using a combination of methods

- with so much existing activity, does it make sense to penetrate this field? (~300 papers/year!)

Example 2: Analysis of prior knowledge and publications

Two types of approaches exist into dynamic strength grading

- correlate measurement method X with measured rupture load
- model the piece of lumber based on measurements and determine the strength based on the model

The latter approach is not very popular

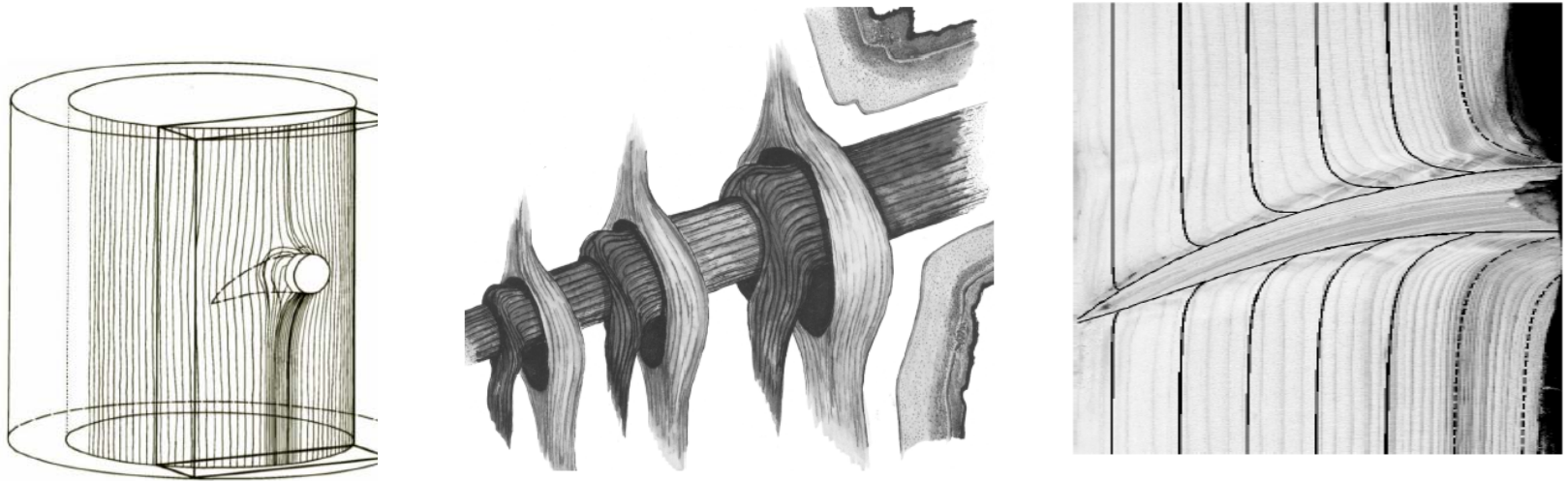
- more mathematical
- has not been better
- most of the researchers have background in wood science and not capable of modeling oriented work

The right question: have the models been a faithful reproduction of the original?

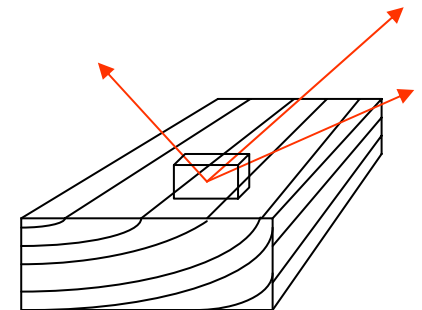
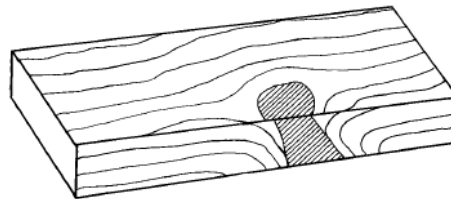
- Not really! E.g., wood grain often modeled as fluid flow as this is directly supported by Finite Element Modeling (FEM) software packages!!!

Example 2: The reality

The fiber direction in wood depends on the knots and their types

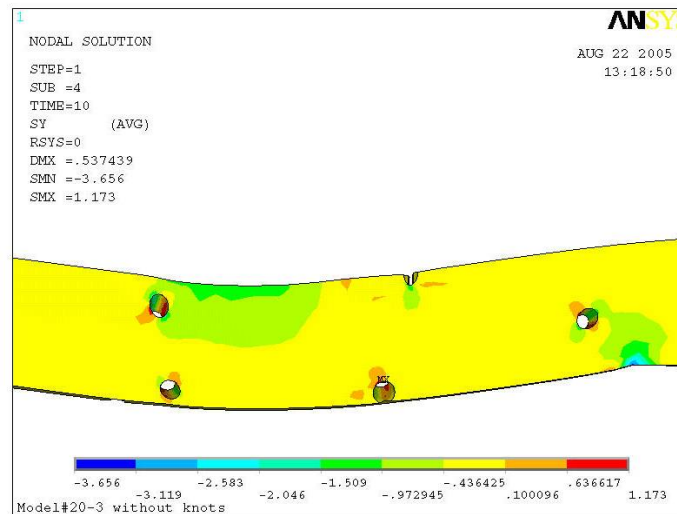


The strenght depends on the structure and differs between axial, tangential and radial directions of grain



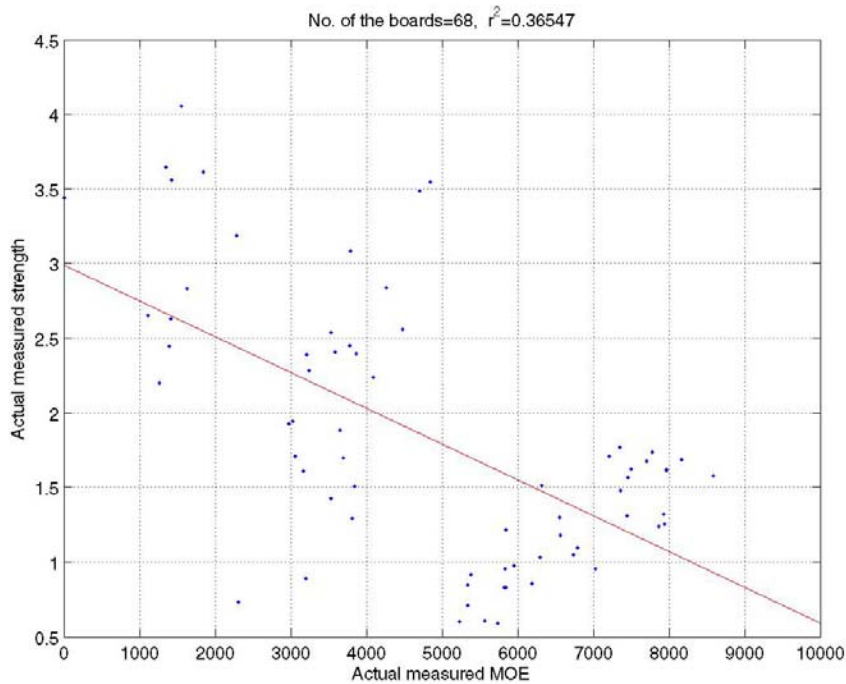
Example 2: The very brief research plan

1. Model the wood faithfully according to knowledge of fiber behavior
2. Determine the strengths using FEM and compare to actual strengths
3. Compare to prior results
4. Ensure industrial utilization and publish; Quickly!

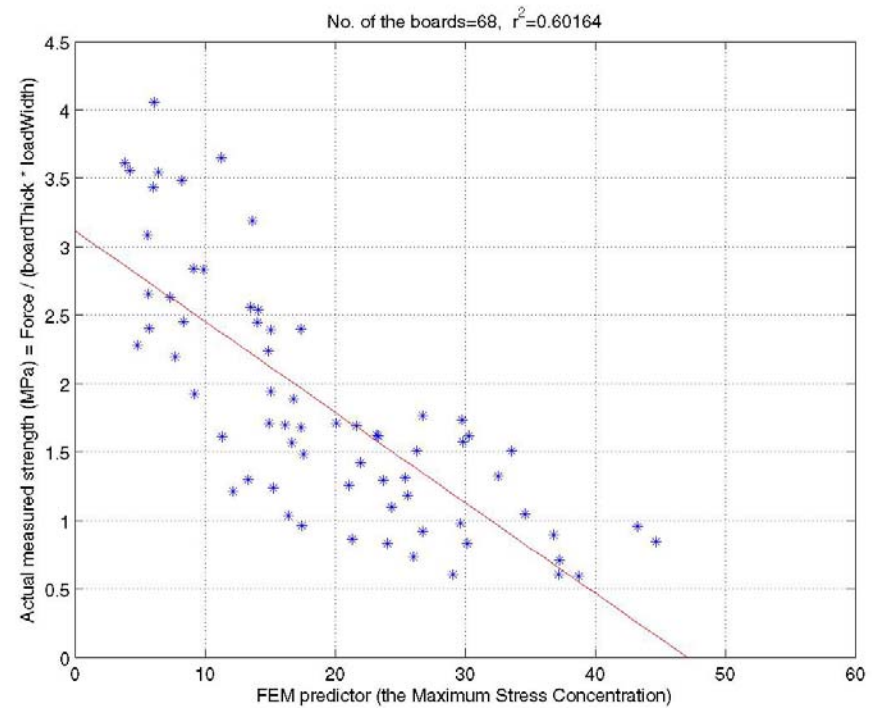


Example 2: The very brief results

Bending test: $r^2 \sim 0.36$



FEM: $r^2 \sim 0.61$



Example 2: What the reviewers are likely to state (hopefully)

The research topic

- Well justified by economic importance and potential to improved human safety

The scientific results

- meticulous modeling of the phenomena captured the essential properties of the material; first time for wood material that is very anisotropic
- 3-D FEM works much better in this problem than 2-D modeling used so far

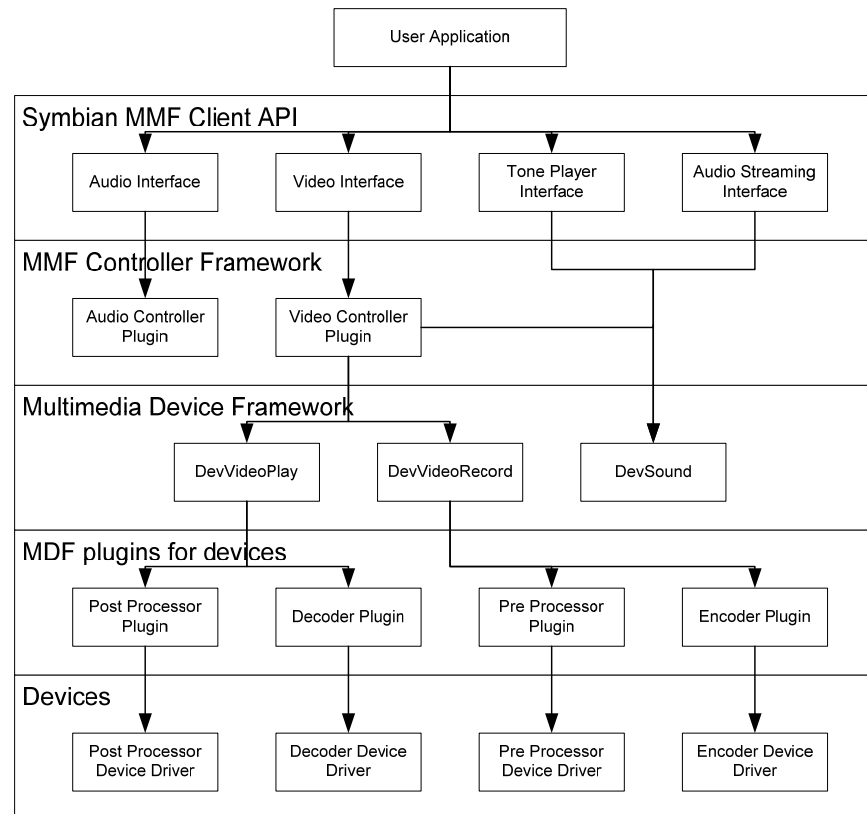
...this is all very interesting, but what has it do with electronics and computer science?

- grading 60-120 boards per minute demands ~100-500GFLOPS by current estimates
- FEM computations in embedded systems: an original contribution!
- Actually, a GPU is used as a computing engine

Example 3: Energy efficient architectures - background (1)

Typical software systems are

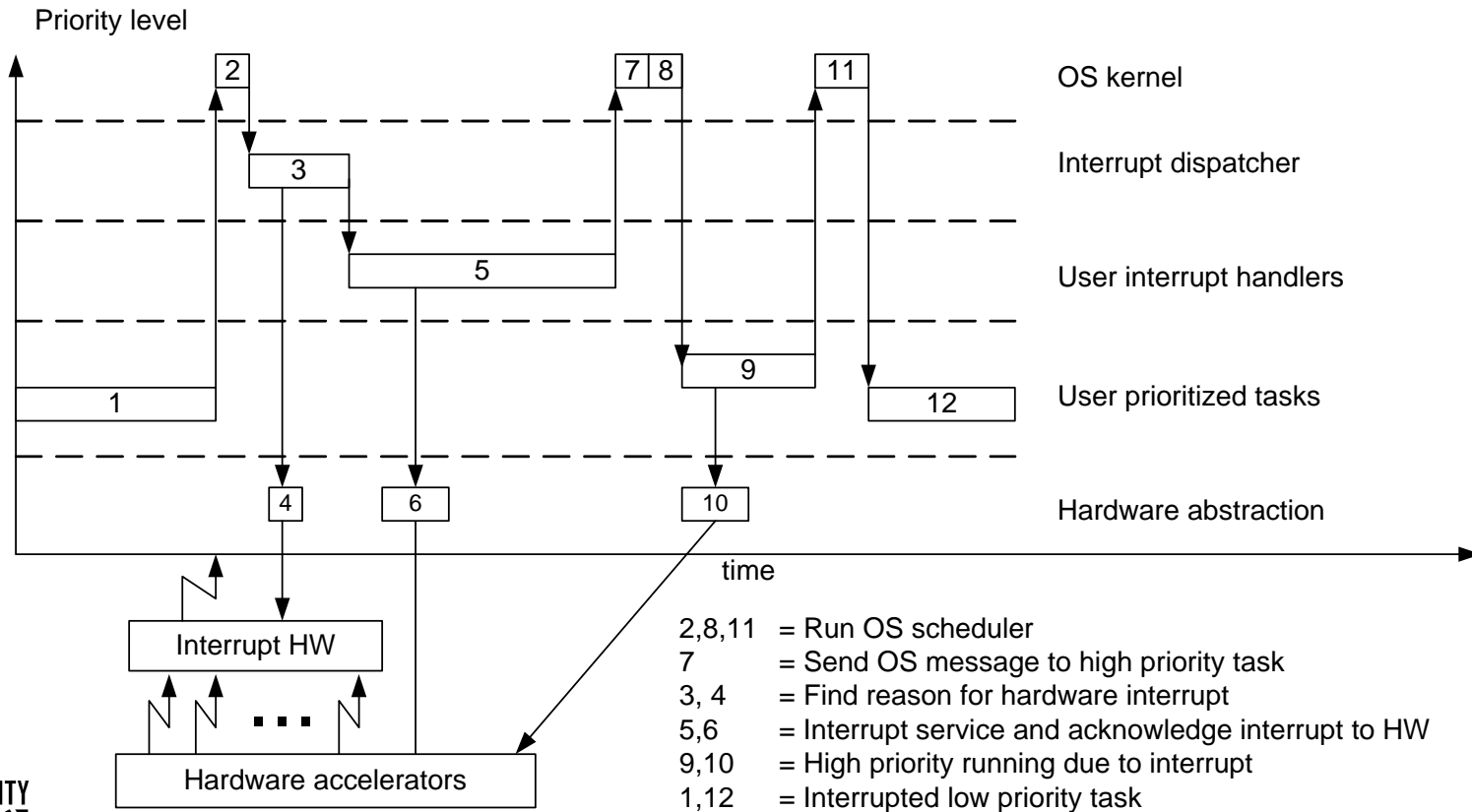
- based on a layered architecture intended to hide details, with
- interrupts for interfacing with hardware, while
- each task has its own context



Example 3: Energy efficient architectures - background (2)

Application, API, and middleware code are fragmented in memory

- Lots of diversions in the execution flow
- cache hit rates can suffer even a single application is executed



Example 3: Energy efficient architectures – exhibit (1)

power efficiency of mobile GSM handsets (basic voice call)

Note: Measurements according to GSMA TW.09

1995:
Nokia 2110
Talk-time 2h40 min
battery 550mAh
~17s/mAh



1998:
Nokia 6110
talk-time 3h
battery 900mAh
~12s/mAh



2003:
Nokia 6600
talk-time 3h
battery 850mAh
~13s/mAh



Characteristics of used silicon technologies

Design rule (nm)	Supply voltage (V)	Approximate normalized power*delay/gate
800 (1995)	5.0	45
500 (1998)	3.3	15
130 (2003)	1.5	1

Example 3: Energy efficient architectures – exhibit (2)

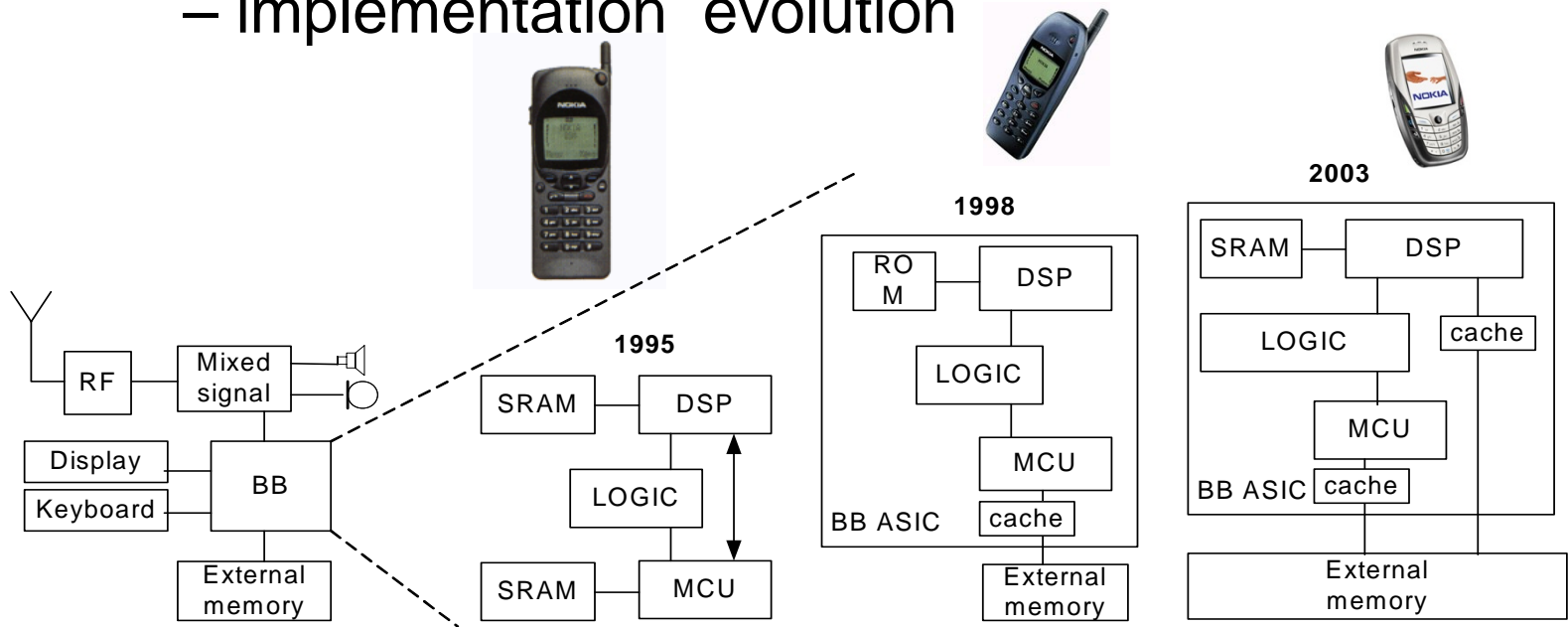
Power efficiencies of a processor family

Processor	Max. clock (MHz)	Power consumption (mW/MHz)	Silicon area (mm ²)
ARM7 (720T)	100	0.2	2.4
ARM9 (926EJ-S)	266	0.45	4.5
ARM10 (1022E)	325	0.6	6.9
ARM11 (1136J-S)	550	0.8	5.55

(130nm CMOS process. Source: ARM foundry program)

Energy per instruction has grown with more recent processor designs

Example 3: Energy efficient architectures – implementation evolution



HW characteristics

HWAs for BB DSP

SoC
Less HWAs

Multilevel memory DSP
More HWAs

SW characteristics

Periodic scheduling
No accelerator interrupts

Middleware
OS

Pre-emptive OSs
Accelerator interrupts

Example 3: Analysis (1) - application efficiency degradations

(approx., GSM voice calls)

The computational cost of voice call implementation	~2.x - ~2.5x
• this is mostly due to voice quality improvements	
Operating system & interrupt overheads	~1.3x - ~1.5x
• RMA alone may require ~1.4x higher clock rate	
API & middleware overheads	~1.2x - ~1.5x
• includes associated cache misses	
Execution time jitter provisioning	~1.3x - ~1.5x
• again due to cache effects	
Processor energy/instruction	~1.5x - ~ 2x
Execution pipeline overheads	>1x
• due to unpredicted branches	
Total	~6x - >10x

Have the degradations due to software solutions now leveled off?

Has anyone noticed this? Yes! But the development considered
as an element of nature...

Example 3: Research ideas (partial list)

How to reduce the overheads from API and middleware layers?

- now they are executable code
- could a compiler be aware of abstraction layers and eliminate code?

How to reduce overheads from hardware/software interfaces?

- interrupts are expensive
- could interrupts be reduced by using multithreading?

How to reduce overheads from scheduling and context switches?

- static scheduling requires ~40% higher clock rates
- could a JIT compiler be used for generating multithreaded tasks?

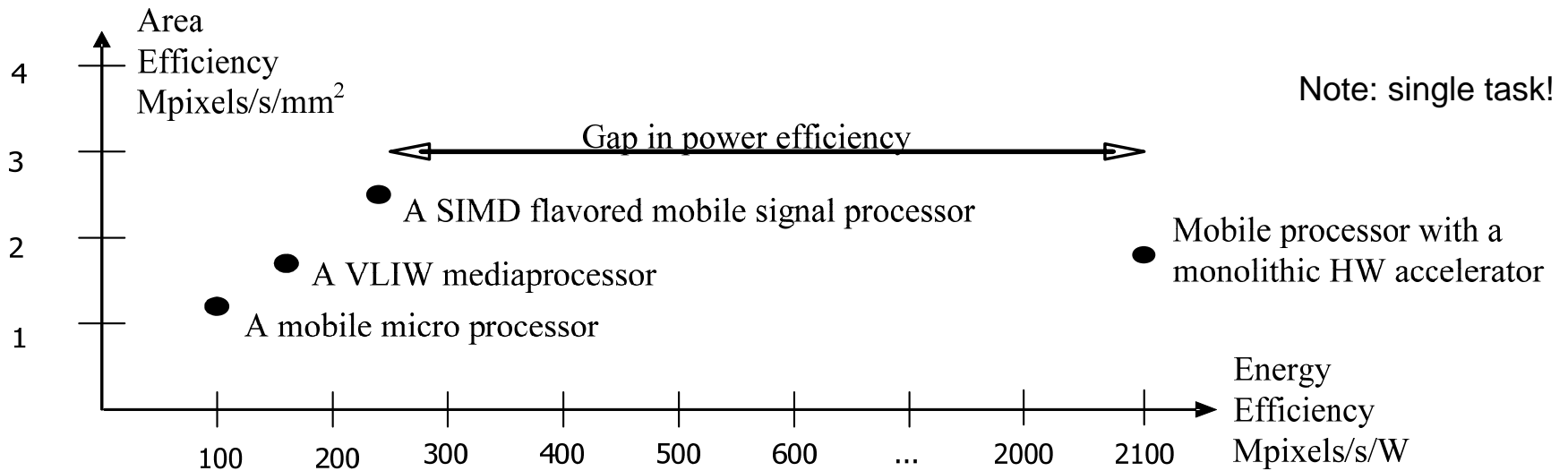
How to achieve flexibility with hardware accelerators?

- monolithic accelerators are inflexible
- fine-grained accelerators are expensive to interface via interrupts
- could a JIT compiler generate multithreaded code that avoids interrupts?

Example 3: Analysis (2) – energy and area efficiencies

Area (Mpixels/s/mm²) and energy (Mpixels/s/W) efficiencies of MPEG-4 encoders

- normalized to a 60nm 1V CMOS process



Hardware acceleration clearly the way to go

Machine Vision Group
Dept. of Electrical and Information Engineering, and Infotech Oulu

Example 3: Hardware accelerators vs. programmable processors

Hardware accelerators are essentially

- efficient dedicated datapath designs for which
- finite state machines are generated for control
- ~5-10pJ/operation (90nm CMOS)

Programmable processors are essentially designed to perform

- instruction fetch and
- execution cycle as fast as possible
- ~125-500pJ/operation (embedded processors)
- ~10-20nJ/operation (general purpose processors)

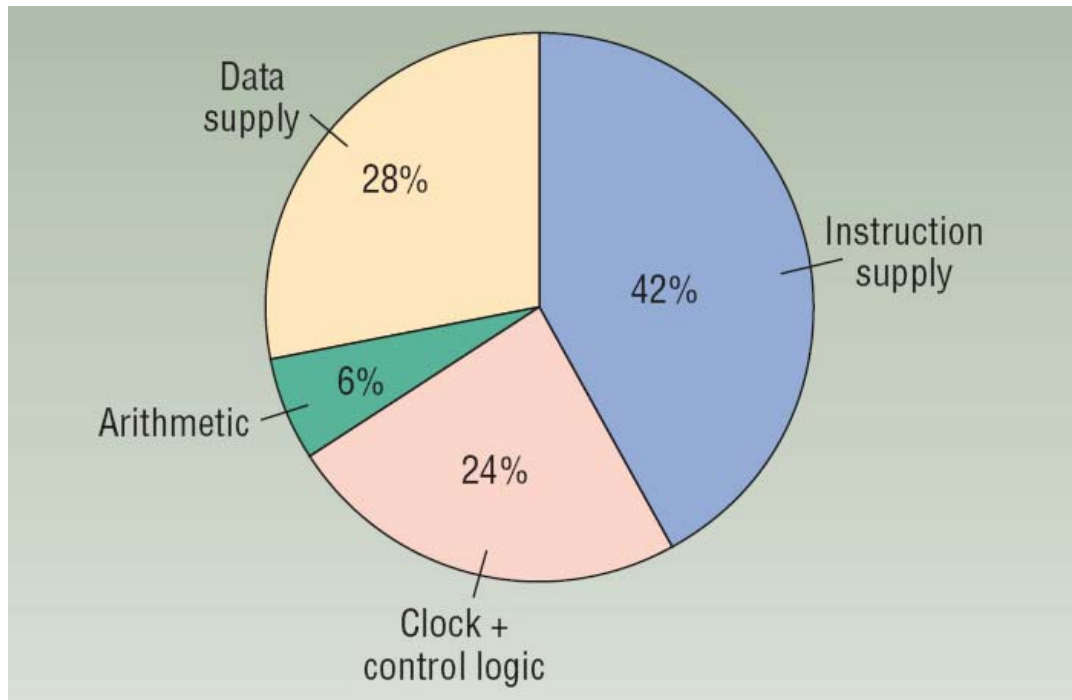
Is it possible to design architectures that combine the

- flexibility of programmability, and
 - the efficiency of the dedicated data paths?
-
- what about high throughput in the same package?

Example 3: Embedded Processor Energy Consumption Breakdown

Notice the small share of arithmetics

- this can be considered as the floor level, however
- tailored word lengths can provide for extra savings in accelerators



(Dally et al
2008)

Example 3: Processing state-of-the-art

Technology/ processor	Gap to ASIC (roughly)	Notes on efficiency	Brief characterization
Desktop processors	200X-	Very expensive I/D supply	GPP; various microarchitectures, hidden bypass network
Embedded application RISC	20X-	Still expensive I/D supply	GPP, hidden bypass network
FlexCore	10X-	Expensive cross-bar	Exposed fully reconfigurable pipeline; GPP
Fixed Point DSP (C55x)	7X-	Depends on <i>application</i>	~ASIP, hidden bypass network
Sandblaster	3X-	<i>SDR</i> , high throughput	Multithreaded, internally exposed pipeline VLIW; GPP/ASIP
Stanford ELM	1.5X-	Low throughput per core	Exposed communications, register files instead of caches
TTA	1X-	Depends on <i>design</i>	ASP architecture template, exposed communications

Example 3: What the reviewers might state from derived research results

The research topic

- aims directly into the heart of computer science and engineering
- very ambitious, multiple PhD thesis effort

The scientific results

- new paradigms for building embedded systems
- significantly improved energy efficiency
- moves the boundary between compilers and abstraction layers

...this is all very interesting, but this isn't what is traditionally done in Finland

- this should have been done much earlier
- the mainline computer science and engineering are not good models for mobile embedded systems

Research being carried out in cooperation: UO, TUT and ÅA

Final words

Finding a Ph.D. research topic is a too important issue to be left for a graduate student

- active supervisor involvement absolutely necessary
- after an initial rough topic fix a written literature review needed:
2-3 landmark papers/books, 10-20 key papers reviewed against the idea

Over 80% of doctoral theses are made in research groups

- if you are not a member of a research group, enlist in one!
- while life of a PhD student with a lone-wolf professor can be exciting, quite few of the students finish their studies in thesis defence

“Chance favors a prepared mind”