How To Give Strong Technical Presentations
Markus Püschel
Department of Computer Science
Biological Fact I: Text Versus Images

Speech
Text

You cannot read and listen at the same time

Ideally

You talking

Verbal channel

You talking

Verbal channel

© 2019 Markus Püschel, Computer Science, ETH Zürich
Handling the Medium

"Teleprompter"

Really bad

"Slideument"

Better

Presentation

Ideal
(not always possible)

Lorem ipsum dolor sit, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Minimize Text
Good presentation slides are not self-contained
Be Clear About the Goal

**Goal 1:** In these 30 minutes explain the entire approach and technology including all relevant details.

**Goal 2:** In these 30 minutes explain what main problem the technology addresses, one or two key ideas in the approach, and one or two key results. Get people excited to learn more.

---

Typical Organization

- **Background (omit if possible)**
- **Results**
  - Motivation
  - Problem statement
  - Your Work
Motivation
Problem statement

- **What? Why? Why important?**
- Exceptionally clear
- If possible, precise problem statement:
  - Given ..., we want to compute ...
  - Input: ..., Output: ....
  - Block diagram showing input/output
- Start interesting
  - example result
  - interesting fact plus source
  - anything that starts the story

**A presentation is a story that starts on the first slide**
The Problem

- Computers architectures have become more complex
  - Memory hierarchies
  - Vector extensions
  - Multiple cores

- Optimizing for software for these features is very difficult
  - Compilers fail to do it
  - Hence the software developer has to do it
  - Requires architecture and algorithm expertise: expensive

- Performance does not port
  - Needs re-optimization for every new processor
  - Without optimization: often 10x performance loss

- Particularly noticeable for computing functions
  - Matrix multiplication
  - Discrete Fourier transform
  - others

Don’t start with a text-only bullet slide
The Problem: Example DFT

Discrete Fourier transform (single precision) on Intel Core i7 (4 cores)
Performance [Gflop/s]

- Same operations count
- Best compiler and flags

Communicate main idea(s) and approach
Do not (try to) communicate every detail of your work

*How to explain technical work well?*
Explaining well

- Visualize
- Use examples not generic explanations
- Small example, full truth

Don’t just talk about it ....

no corners red
\chi^2 + \psi^2 = \chi^2
... show it!

Statistical Classification: C4.5

- C4.5 generates decision trees from training data
- The trees can be used for classification
- Formally:
  - Input: Training set of size $m$; each member has $n$ features
  - Output: decision trees mapping samples to classes

Example: not so good

```
<table>
<thead>
<tr>
<th>Features</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{1,1}$</td>
<td>$C_1$</td>
</tr>
<tr>
<td>$x_{1,2}$</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$x_{m,1}$</td>
<td></td>
</tr>
<tr>
<td>$x_{m,2}$</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$x_{m,n}$</td>
<td>$C_m$</td>
</tr>
</tbody>
</table>
```

- Output: decision trees mapping samples to classes
### Statistical Classification: C4.5

**Outlook** | **Temperature** | **Humidity** | **Windy** | **Decision**
---|---|---|---|---
Sunny | 85 | 85 | False | Don’t play
Sunny | 80 | 90 | True | Play
Overcast | 83 | 78 | False | Play
Rain | 70 | 96 | False | Don’t play
Rain | 68 | 80 | False | Don’t play
Rain | 65 | 70 | True | Don’t play
Overcast | 64 | 65 | True | Play
Sunny | 72 | 95 | False | Play
Rain | 75 | 80 | False | Play
Sunny | 75 | 70 | True | Play
Overcast | 72 | 90 | True | Play
Rain | 81 | 75 | False | Don’t play
Rain | 71 | 80 | True | Don’t play

\[
P(\text{play}|\text{windy} = \text{false}) = \frac{6}{8}
\]
\[
P(\text{don’t play}|\text{windy} = \text{false}) = \frac{2}{8}
\]
\[
P(\text{play}|\text{windy} = \text{true}) = \frac{1}{2}
\]
\[
P(\text{don’t play}|\text{windy} = \text{true}) = \frac{1}{2}
\]

\[
H(\text{windy} = \text{false}) = 0.81
\]
\[
H(\text{windy} = \text{true}) = 1.0
\]
\[
H(\text{windy}) = 0.89
\]
\[
H(\text{outlook}) = 0.69
\]
\[
H(\text{humidity}) = ...
\]

### Linear Transforms

\[
\begin{pmatrix}
y_0 \\
y_1 \\
\vdots \\
y_{n-1}
\end{pmatrix} = y = Tx
\]

\[
x = \begin{pmatrix}
x_0 \\
x_1 \\
\vdots \\
x_{n-1}
\end{pmatrix}
\]

**Example:**

\[
T = \text{DFT}_n = [e^{-2\pi i k\ell/n}]_{0 \leq k, \ell \leq n}
\]
Fast Fourier Transforms (FFTs)

- Can be expressed as structured matrix factorizations
  \[ \text{DFT}_{mn} = (\text{DFT}_m \otimes I_n)T_{mn}^{mn}(I_m \otimes \text{DFT}_n)L_{mn}^{mn} \]

- Formalism:
  - \( L_{mn} \) \( i + j \mapsto jm + i, \ 0 \leq i < n, \ 0 \leq j < m \)
  - \( I_n \) \( n \times n \) identity matrix
  - \( A \otimes B \) \( [a_{k,l}]_{0 \leq k, l < n}, \ A = [a_{k,l}] \)
  - \( T_{mn} \) a diagonal matrix

Fast Fourier Transform: Size 4

\[
\begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & i & -1 & -i \\
1 & -1 & 1 & -1 \\
1 & -i & -1 & i \\
\end{bmatrix}
\begin{bmatrix}
x \\
\end{bmatrix}
= 
\begin{bmatrix}
1 & -1 & 1 & -1 \\
1 & -i & -1 & i \\
1 & -1 & -1 & i \\
1 & 1 & -1 & 1 \\
\end{bmatrix}
\begin{bmatrix}
x \\
\end{bmatrix}
\]

12 adds, 4 mults  4 adds  1 mult  4 adds

Matrix formalism:
\[ \text{DFT}_4 = (\text{DFT}_2 \otimes I_2)T_2^4(I_2 \otimes \text{DFT}_2)L_2^4 \]
Other Transform Algorithm

Example: good (then full truth)

Precise experimental setup

More later on data
Common Mistakes

- Thinking: If one can understand it well, people will think it’s trivial
- Too many slides
- Slides too packed

External Material

- Everything included with copy-paste:
  Images, graphics, text (even if only one sentence)

  Acknowledge source on the same slide!
  bottom right, gray is one option
Design is about efficient communication, not about making things pretty
Two Design principles

- Alignment
- Layering

Alignment

- Everything is aligned to something else
- If in doubt align *left*
Nervousness

- Practice the presentation
- Be perfectly prepared
- Take every small opportunity to present
- If it's really bad
  - Try some tricks from books
  - See a specialist

**Top 10 fears**
1. Fear of snakes
2. Fear of public speaking
3. Fear of heights
4. Fear of closed spaces
5. Fear of spiders (and insects)
6. Fear of needles
7. Fear of mice
8. Fear of flying
9. Fear of dogs
10. Fear of thunder
11. Fear of crowds

Source: U.S.A. Gallup Poll, February 18-21, 2001 (1,016 respondents)

Layering

Hierarchical organization of elements through proper use of contrast, emphasis, and de-emphasis
Nervousness

Top 10 fears
1. Fear of snakes
2. Fear of public speaking
3. Fear of heights
4. Fear of closed spaces
5. Fear of spiders (and insects)
6. Fear of needles
7. Fear of mice
8. Fear of flying
9. Fear of dogs
10. Fear of thunder
11. Fear of crowds

Source: U.S.A. Gallup Poll, February 18-21, 2001 (1,016 respondents)

- Practice the presentation
- Be perfectly prepared
- Take every small opportunity to present
- If it’s really bad
  - Try some tricks from books
  - See a specialist

Layering

Top 10 fears
1. Fear of snakes
2. Fear of public speaking
3. Fear of heights
4. Fear of closed spaces
5. Fear of spiders (and insects)
6. Fear of needles
7. Fear of mice
8. Fear of flying
9. Fear of dogs
10. Fear of thunder
11. Fear of crowds

Source: U.S.A. Gallup Poll, February 18-21, 2001 (1,016 respondents)
Nervousness

Top 10 fears
1. Fear of snakes
2. Fear of public speaking
3. Fear of heights
4. Fear of closed spaces
5. Fear of spiders (and insects)
6. Fear of needles
7. Fear of mice
8. Fear of flying
9. Fear of dogs
9. Fear of thunder
9. Fear of crowds
Source: U.S.A. Gallup Poll, February 18–21, 2001
(1,016 respondents)

- Practice the presentation
- Be perfectly prepared
- Take every small opportunity to present
- If it’s really bad
  - Try some tricks from books
  - See a specialist

P6 Memory System

![Diagram of P6 Memory System]
P6 Memory System

- DRAM
- External system bus (e.g. PCI)
- L2 cache
- L1 I-cache
- Instruction fetch unit
- Instruction TLB
- Data TLB
- L1 D-cache
- Cache bus
- Processor package

Example good
Code

```c
#include <ia32intrin.h>

// n a multiple of 4, x is 16-byte aligned
void addindex_vec(float *x, int n) {
    __m128 index, x_vec;
    for (int i = 0; i < n/4; i++) {
        x_vec = _mm_load_ps(x+i*4); // load 4 floats
        index = _mm_set_ps(i+3, i+2, i+1, i); // create vector with indexes as values
        x_vec = _mm_add_ps(x_vec, index); // add the two
        _mm_store_ps(x+i*4, x_vec); // store back
    }
}
```

- Fixed-width font (Consolas or Courier)
- Layering for readability

Presenting a Viewgraph: Example

- Start like this:
  - We compare the performance of Spiral and IPP
  - The x-axis shows ...., the y-axis shows
  - This means higher is better (or vice-versa)
  - For example, this datapoint means that ....
- Now you can explain more
- Then conclude
- But this plot is bad...
DFT $2^n$: Performance on Pentium 4, 2.53 GHz

- Spiral SSE
- Intel MKL
- Spiral C (vectorized)
- Spiral C

Horizontal y-label

Attractive font (sans serif, here: Gill Sans)

Main line emphasized (red, thicker)

Background/grid inverted for better layering

No y-axis (superfluous)

No legend for faster decoding

Final words
Creating a Presentation

- Think about good visuals (diagrams, graphs, fotos, screenshots) to support the story; then sketch the presentation on paper

How to Get Better

- Study the principles and apply them
- Give your best in every presentation
- Learn to verbalize the reason for design decisions and for problems with a slide
  - Explain and help others
  - Evaluate presentations you see
- Reduce text more and more
- Think hard about visualizations and good examples
- Experiment
- Expand your knowledge
  - Books (next slide)
  - Watch great presentations online (e.g., TED talks)
Last Tip: Never end with a slide

Thank you!

Some Books This Lecture Draws From

- Cliff Atkinson, *Beyond Bullet Points*, Microsoft Press, 2005
- Stephen Few, *Show Me the Numbers*, Analytics Press, 2004