COS700

Lecture 2: Research Proposal

Research Proposal

- See “Research Proposal Instructions” document on COS 700 website – to be added (will announce)
- Due date: **1 May 2017**
- Submit using Easychair system
- Peer review (due **31 May 2017**)
- Will use Latex

Research Proposal

- The overall idea: Expresses in formal writing
  - What you are going to do
  - Why you are going to do it, and
  - How you are going to do it
- If done properly, it constitutes a significant amount of the work of the overall research project
- Is it really necessary to write a proposal?
  - A weak proposal is a waste of time
  - A strong proposal will save you time

Research Proposal (cont)

- Advantages:
  - An agreement between you and your supervisor
  - Forces you to think clearly and precisely about the research before you do too much of the work
  - Forces you to start writing early
  - Many of the paragraphs should end up in the final report, so not wasted effort

Typical Structure

Prefix:

- Page 1: Cover page with essential information
- Page 2: Title repeated with abstract and keywords

Proposal:

1. Introduction
2. Problem statement
3. Literature survey
4. Methodology
5. Planning
6. References

Slight deviations fine, but basic elements must be there

Cover Page

- The words “University of Pretoria”, “Computer Science Department”, “COS700 Research Proposal”
- The date (at least the month and year)
- Title of research project
- Student name, number, email address:
  - Should usually be there, but **not** for anonymous peer review
  - **Only** include student number for now
- Name of supervisor(s)
- Will provide **Latex** template

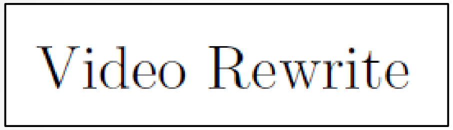
©Ron Leishman * illustrationsOf.com/1122060

Title

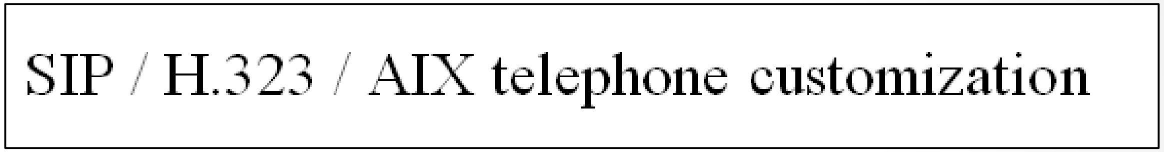
What do you think makes a title a ***good*** *title?*

Real Examples of CS Research Project Titles

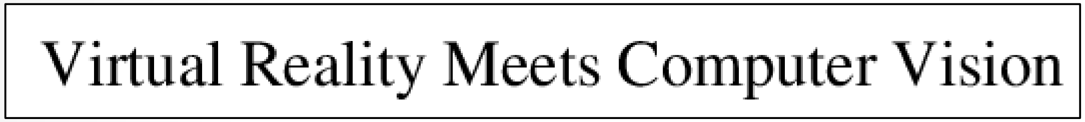
Are these good titles? Why / why not?



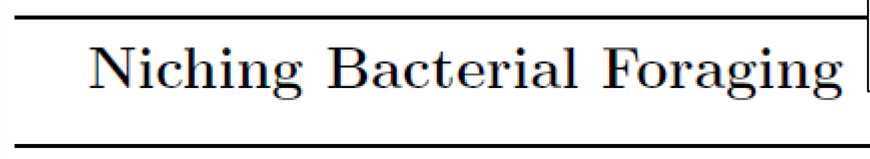
Video Rewrite



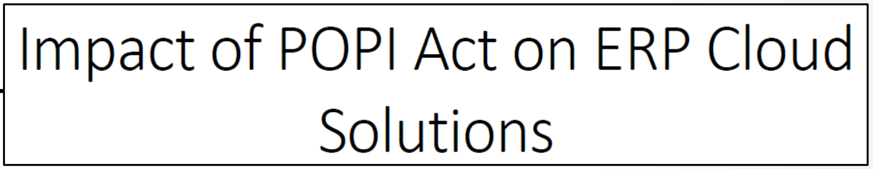
SIP / H.323 / AIX telephone customization



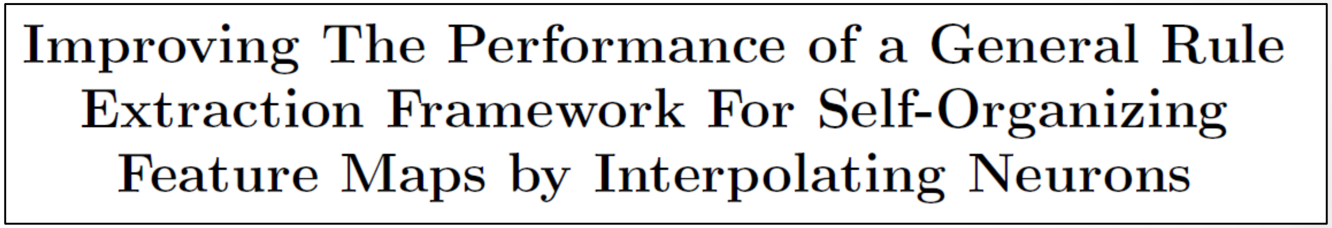
Virtual Reality Meets Computer Vision



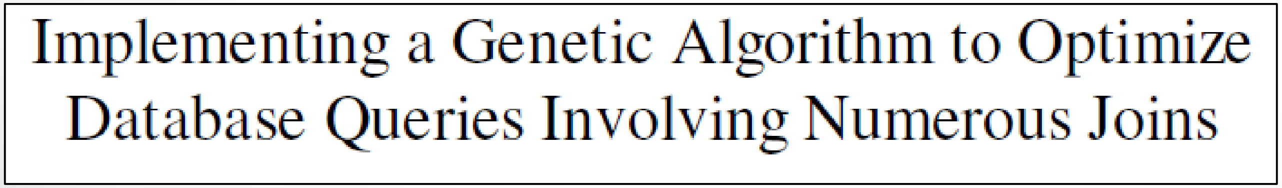
Niching Bacterial Foraging



Impact of POPI Act on ERP Cloud Solutions



Improving The Performance of a General Rule Extraction Framework For Self-Organizing Feature Maps by Interpolating Neurons



Implementing a Genetic Algorithm to Optimize Database Queries Involving Numerous Joins

Title

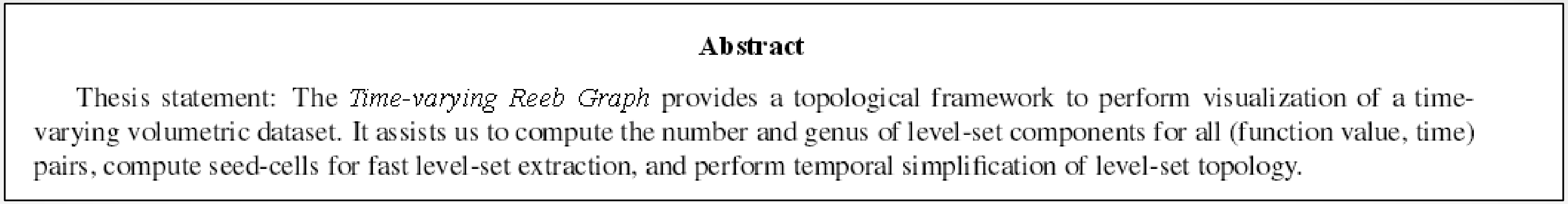
A good title:

- *Captures* the essence of the research project, i.e. the topic of the research
- Must be *specific* (enough detail), but *concise* (no unnecessary words)
- Must (usually) not contain abbreviations or acronyms

Abstract

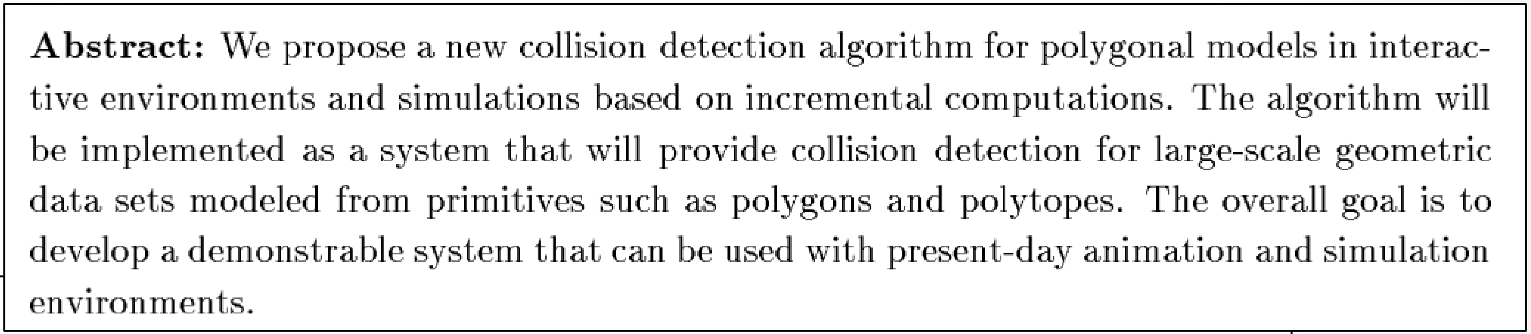
- Why do you think it is called an abstract?
- Length: 100 – 200 words
- Content:
  - Brief background to the research (no citations) & problem statement
  - Method(s) to be used
  - Results
- Usually easier to write last
- Do not just cut and paste from the proposal: write in a more condensed form

Example Project Abstracts

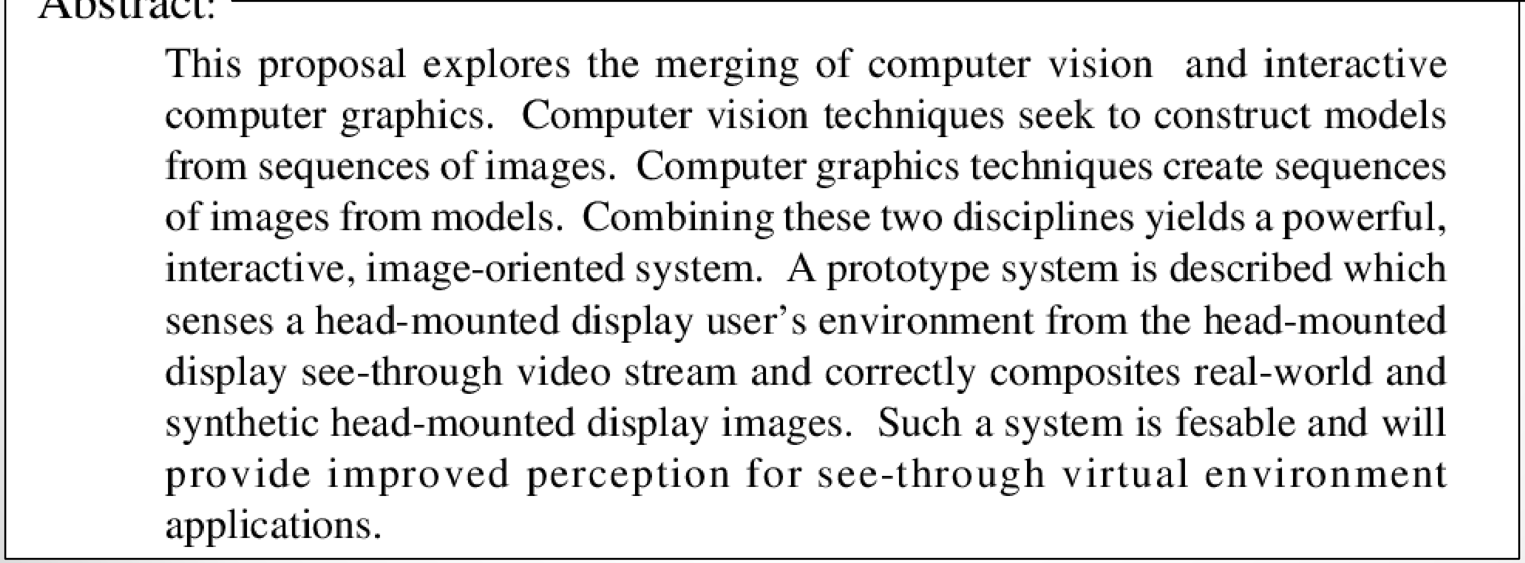


Abstract

Thesis statement: The *Time-varying Reeb Graph* provides a topological framework to perform visualization of a time-varying volumetric dataset. It assists us to compute the number and genus of level-set components for all (function value, time) pairs, compute seed-cells for fast level-set extraction, and perform temporal simplification of level-set topology.



Abstract: We propose a new collision detection algorithm for polygonal models in interactive environments and simulations based on incremental computations. The algorithm will be implemented as a system that will provide collision detection for large-scale geometric data sets modeled from primitives such as polygons and polytopes. The overall goal is to develop a demonstrable system that can be used with present-day animation and simulation environments.

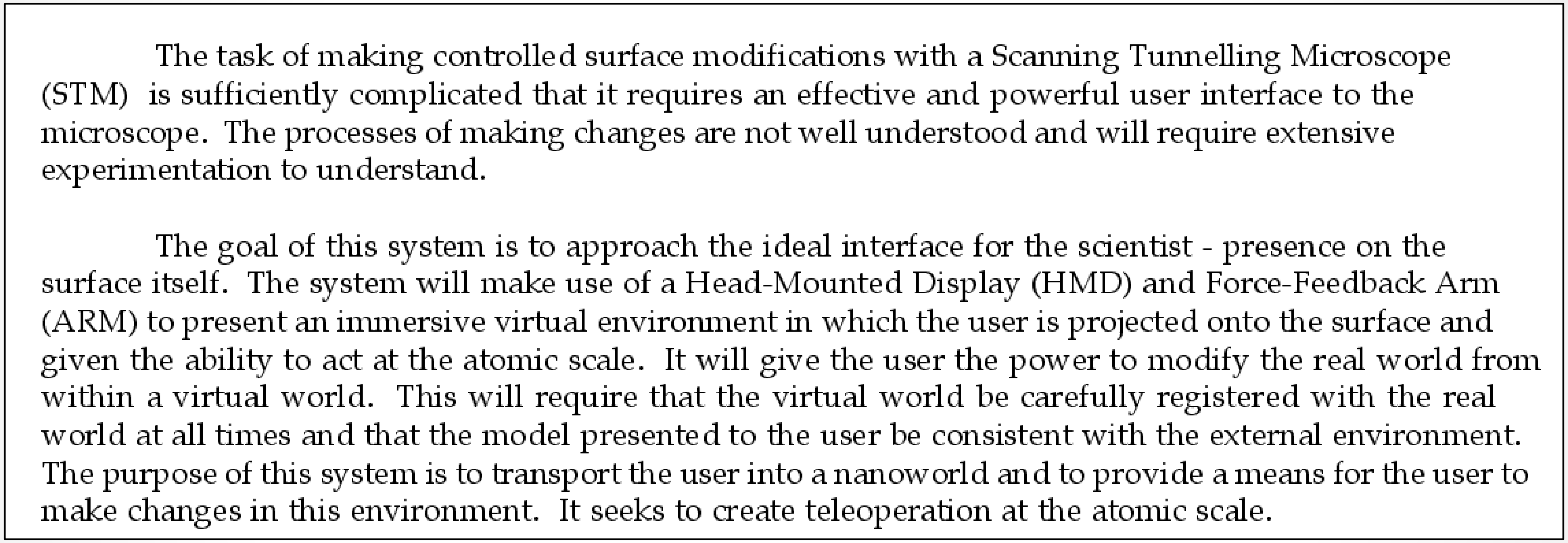


Abstract:

This proposal explores the merging of computer vision and interactive computer graphics. Computer vision techniques seek to construct models from sequences of images. Computer graphics techniques create sequences of images from models. Combining these two disciplines yields a powerful, interactive, image-oriented system. A prototype system is described which senses a head-mounted display user's environment from the head-mounted display see-through video stream and correctly composites real-world and synthetic head-mounted display images. Such a system is fesable and will provide improved perception for see-through virtual environment applications.

Are they any good?

Another Example Abstract



The task of making controlled surface modifications with a Scanning Tunnelling Microscope (STM) is sufficiently complicated that it requires an effective and powerful user interface to the microscope. The processes of making changes are not well understood and will require extensive experimentation to understand.

The goal of this system is to approach the ideal interface for the scientist - presence on the surface itself. The system will make use of a Head-Mounted Display (HMD) and Force-Feedback Arm (ARM) to present an immersive virtual environment in which the user is projected onto the surface and given the ability to act at the atomic scale. It will give the user the power to modify the real world from within a virtual world. This will require that the virtual world be carefully registered with the real world at all times and that the model presented to the user be consistent with the external environment. The purpose of this system is to transport the user into a nanoworld and to provide a means for the user to make changes in this environment. It seeks to create teleoperation at the atomic scale.

Exercise

Abstract Examples

Introduction

- Introduce the background against which your research will be conducted
- Aim is to convince the reader that the research:
  - Is relevant, necessary and timely
  - Has not been done before
- What to put in the introduction:
  - Start with broad field into which your research falls
  - Become more specific
  - Lead into the problem statement
  - Use references throughout (but it is not the full literature survey)
  - Layout of rest of proposal

Introduction (cont)

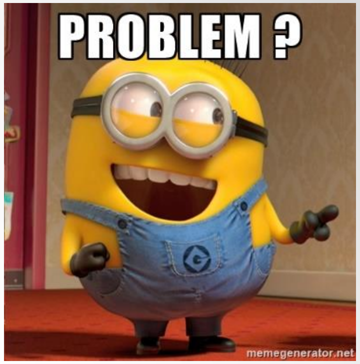
Example project: Multicasting of multimedia via a metropolitan area network

- Could start with a statement about how the use of multimedia data has increased
- Then move onto the high bandwidth requirements of multimedia data and some ways multimedia data is communicated
- Outline the shortcomings of existing approaches
- Introduce multicasting & metropolitan area networks
- Summarize some of the relevant existing work

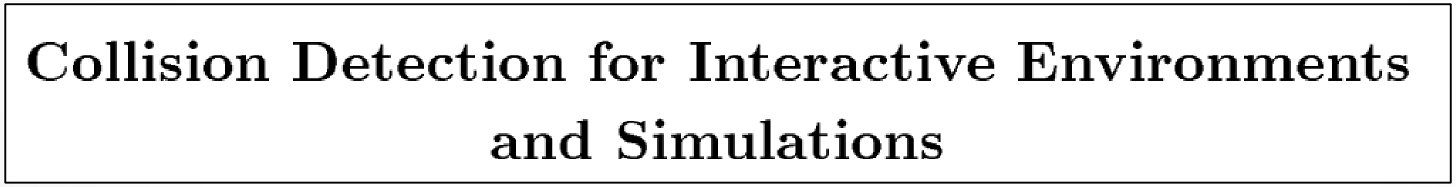
Exercise

Introduction Examples

Problem Statement



- Describe the problem that your research will address in as much detail as possible
- Stating your title as a question sometimes helps in naming the problem

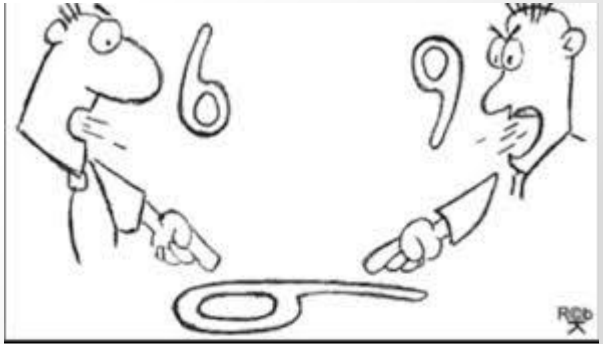


Collision Detection for Interactive Environments and Simulations

How will you rephrase this title as a question?

Problem Statement (cont)

- Specify and describe your main objectives and sub-objectives:
  - to demonstrate …, to evaluate…, to determine…, to establish…, to argue…, to prove …, etc
- Define the scope and limitations of the research
  - What your work deals with and what it does not deal with
  - "This work does not consider…"
- State any assumptions that you are aware of



Exercise

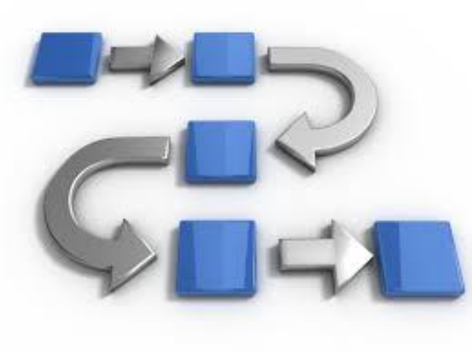
What is the problem statement in the Examples handed out in class?

Can you find it in the abstract or introduction?

Literature Survey

- Overview of research that has been conducted on the problem that you are investigating
- Only use published references
- Work should be directly linked to your work (and you must point out the link)
- Want to show how your research relates to and differs from existing research
- **NB**: Identify the main players in the field
- Want to tell a story – must flow

Methodology



- Identify the methods that you plan to use to address your objectives:
  - Which paradigms and tools?
  - What code do you expect to write?
  - What experiments will you need to conduct? (briefly outline)
- Must tie back to your research objectives
- Should be clear that selected methods are appropriate (explain why alternative methods not proposed)

Project planning

- Identify and describe the specific phases of your work
  - E.g. reading, writing, coding, experimentation, testing, writing, editing, writing
- Provide a schedule of everything that needs to be done all the way up to the submission and presentation date:
  - Tasks that need to be done and the time allocated to them with deadlines
  - Remember that your supervisor needs time to review your report/paper, and Y
  - You need time to incorporate changes before the submission date

Project planning (cont)

Advantages of task timelines:

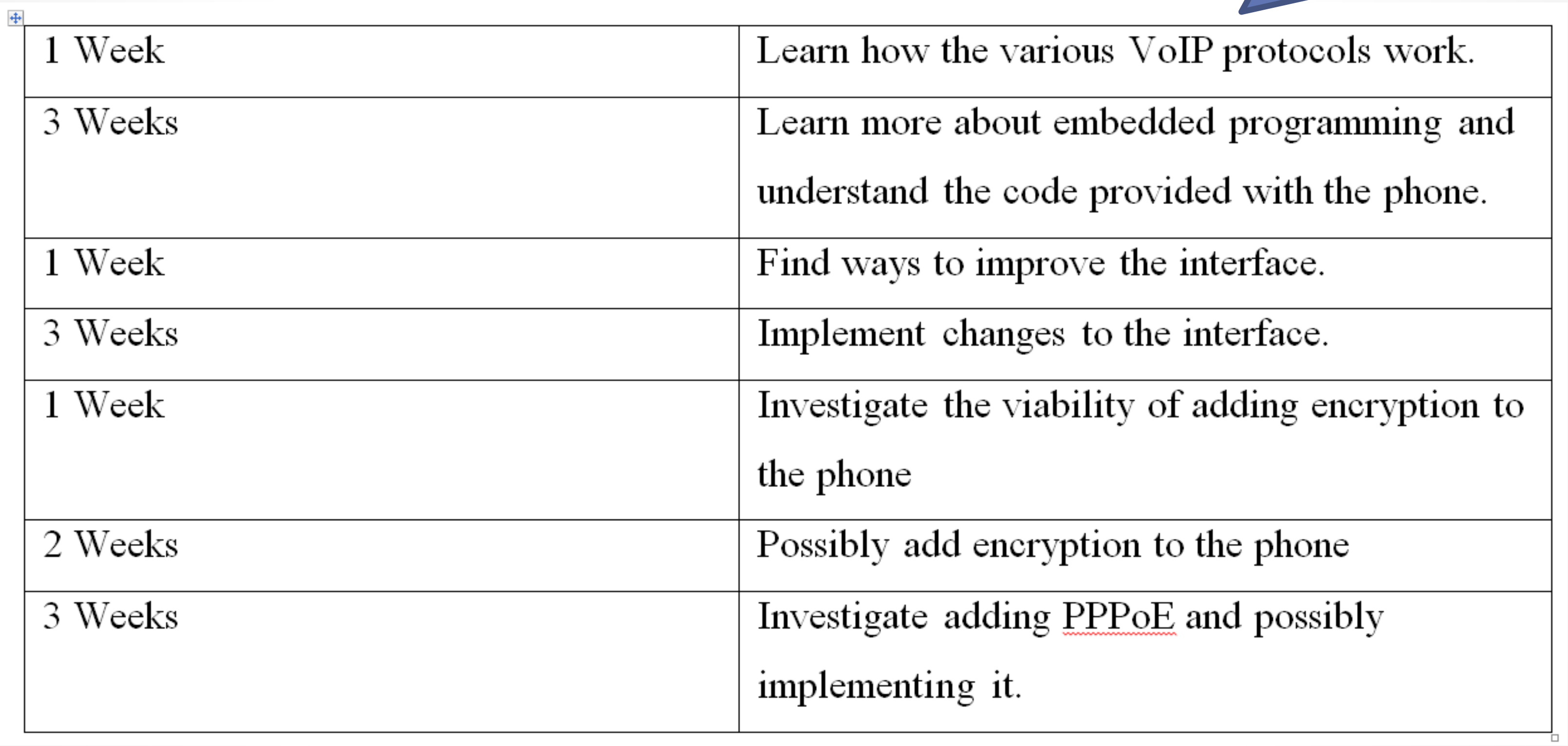
- Forces you to break down the work into manageable chunks
- Lets you think about the logical sequence of tasks
- Planning helps you to not miss out important tasks
- Creates deadlines for yourself
- Can be used to map progress

Schedule – Some Hints

- Be realistic, rather pad the time than set unrealistic deadlines
- Don't forget to factor in other things (other modules, your life, …)
- Some tasks can overlap in time – it's good to have alternative tasks to work on
- Give page estimates to your writing tasks
- Start from the end and plan backwards
- Commit to your plan: have your timeline visible somewhere
- Update your plan if necessary as you progress with the work

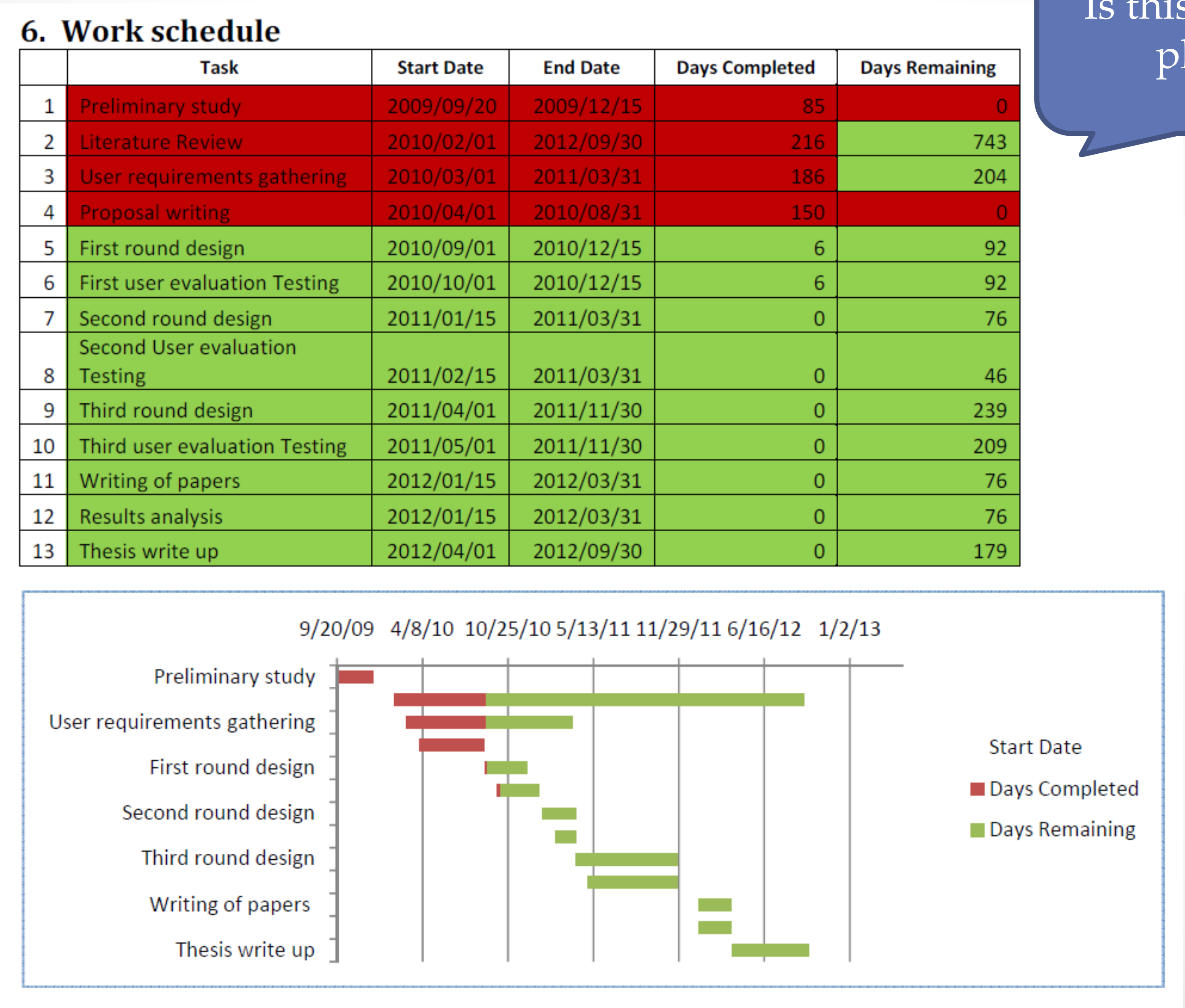
Example 1

Is this a good plan?



1 Week	Learn how the various VoIP protocols work.
3 Weeks	Learn more about embedded programming and understand the code provided with the phone.
1 Week	Find ways to improve the interface.
3 Weeks	Implement changes to the interface.
1 Week	Investigate the viability of adding encryption to the phone
2 Weeks	Possibly add encryption to the phone
3 Weeks	Investigate adding PPPoE and possibly implementing it.

Example 2



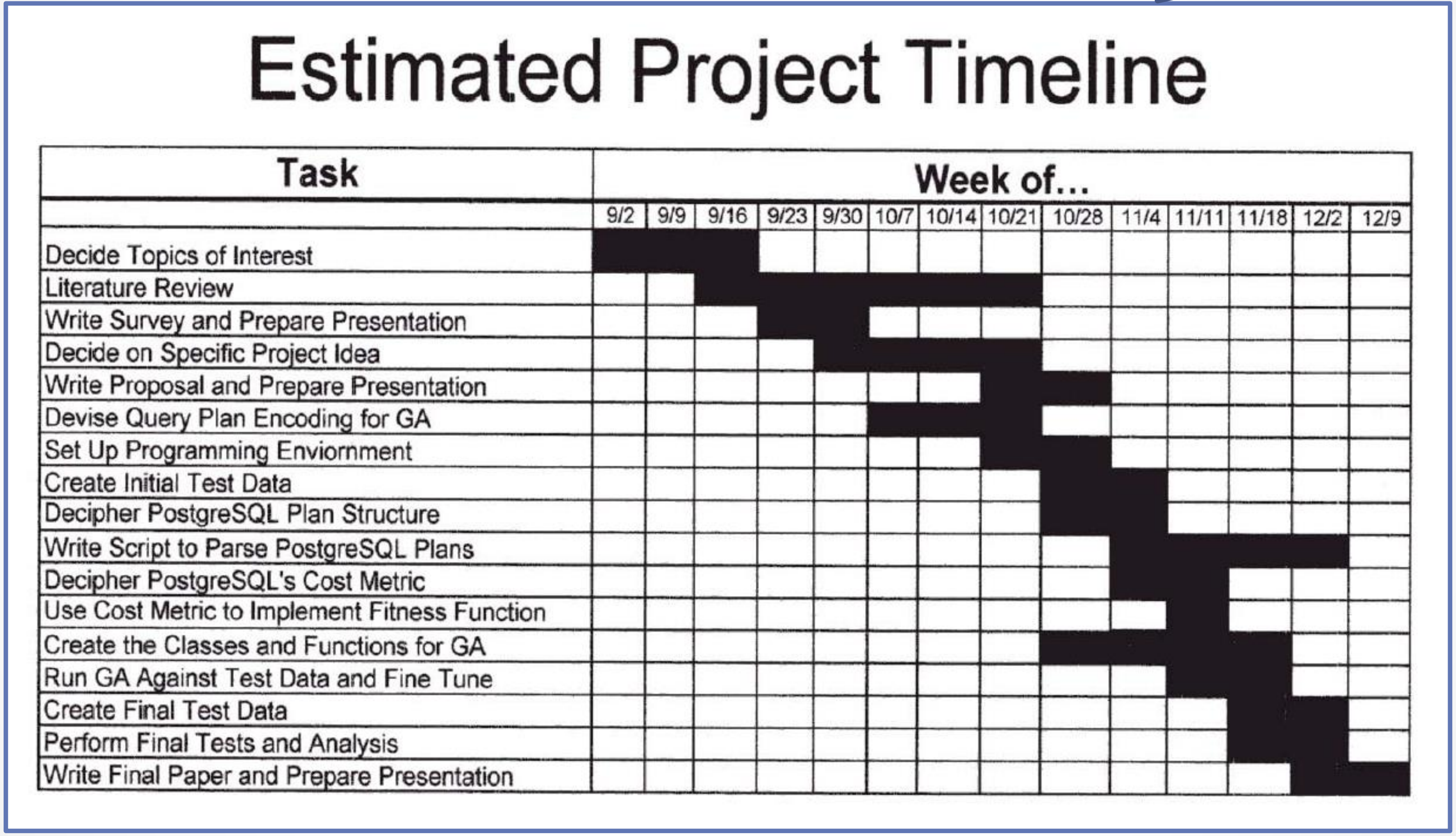
6. Work schedule

	Task	Start Date	End Date	Days Completed	Days Remaining
1	Preliminary study	2009/09/20	2009/12/15	85	0
2	Literature Review	2010/02/01	2012/09/30	216	743
3	User requirements gathering	2010/03/01	2011/03/31	186	204
4	Proposal writing	2010/04/01	2010/08/31	150	0
5	First round design	2010/09/01	2010/12/15	6	92
6	First user evaluation Testing	2010/10/01	2010/12/15	6	92
7	Second round design	2011/01/15	2011/03/31	0	76
8	Second User evaluation Testing	2011/02/15	2011/03/31	0	46
9	Third round design	2011/04/01	2011/11/30	0	239
10	Third user evaluation Testing	2011/05/01	2011/11/30	0	209
11	Writing of papers	2012/01/15	2012/03/31	0	76
12	Results analysis	2012/01/15	2012/03/31	0	76
13	Thesis write up	2012/04/01	2012/09/30	0	179

Is this a good plan?

Example 3

Is this a good plan?

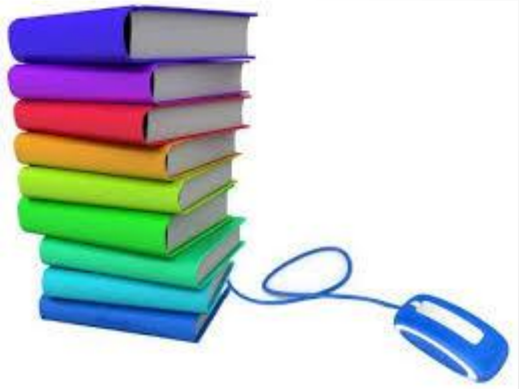


Estimated Project Timeline

Task	Week of...													
	9/2	9/9	9/16	9/23	9/30	10/7	10/14	10/21	10/28	11/4	11/11	11/18	12/2	12/9
Decide Topics of Interest	■	■	■											
Literature Review			■	■	■	■	■	■						
Write Survey and Prepare Presentation				■	■									
Decide on Specific Project Idea					■	■	■	■						
Write Proposal and Prepare Presentation								■	■					
Devise Query Plan Encoding for GA						■	■	■						
Set Up Programming Enviornment								■	■					
Create Initial Test Data									■	■				
Decipher PostgreSQL Plan Structure									■	■				
Write Script to Parse PostgreSQL Plans										■	■	■	■	
Decipher PostgreSQL's Cost Metric										■	■			
Use Cost Metric to Implement Fitness Function											■			
Create the Classes and Functions for GA									■	■	■	■		
Run GA Against Test Data and Fine Tune											■	■		
Create Final Test Data												■	■	
Perform Final Tests and Analysis												■	■	
Write Final Paper and Prepare Presentation													■	■

References

- Provide a list of references that you have cited in the proposal



Academic Writing – Some Tips

- Do not try to impress by sounding ‘academic’
- Simple writing is more effective than complicated writing
- You cannot be clear if you do not know what you are trying to say
- Avoid unnecessary jargon & acronyms
- Define all abbreviations before using them

Academic Writing – Some Tips

Academic writing is formal:

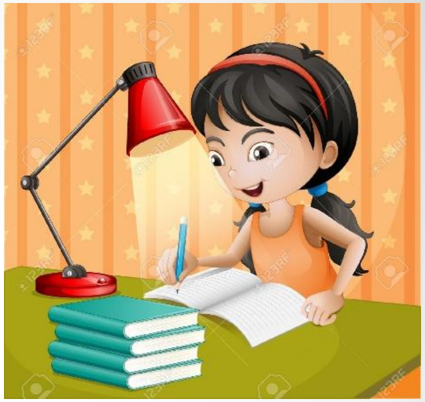
- Don’t use contractions, e.g. couldn’t => could not
- Avoid “really”, “very” “a lot”:
  - E.g. “A lot of time is wasted” -> “Time is wasted”
  - E.g. “A lot of the students” -> “Many students”
- Rather avoid 1st person: “We show that …” vs “The research shows that …” (ask your supervisor)

Academic Writing – Some Tips

- If you can, rather use active voice than passive voice:
  - E.g. "The approach used by Smith et al. [12] to solve the problem was to implement a mock interface." = >

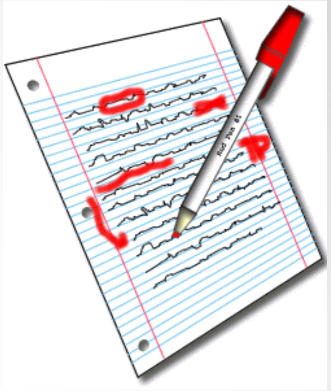
    "Smith et al. [12] implemented a mock interface to solve the problem."
  - E.g. "In section 4 the experiments are discussed." => "Section 4 discusses the experiments."
- Writing should flow:
  - Introduce sections, end sections
  - Reader should know what they are reading and why

Academic Writing – Some Tips



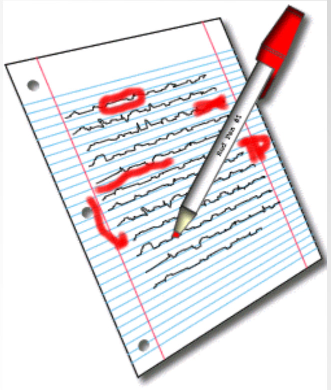
- Be careful of words that are absolutes, e.g.:
    - "never", "perfectly", "completely"
    - "proves" – are you really proving it?
- Be careful of "it", "this", "these", unless there is absolutely no ambiguity
- Past tense or present tense?
    - Work previously done usually in past tense e.g. "parameters were set to …", literature review
    - Results usually in past tense
    - The rest in present tense ("This paper proposes …").
    - Conclusions also usually in past tense: "A model was proposed … it was shown that …"

Academic Writing – Some Tips



- Take the time to read your own writing
- Cut unnecessary words:
  - Avoid “There is / There are”. E.g. “There are many ways in which the travelling salesman problem can be solved.” => “The travelling salesman problem can be solved in many ways.”
  - Other phrases to avoid: “As it is well known”, “It can be regarded that”, “It should be regarded that”, “It is important to note that”

Academic Writing – Some Tips



- Cut unnecessary words:
  - See video: "**Writing in the Sciences – Cut the Clutter**", Stanford School of Medicine (YouTube)
  - Refer to COS700 Text book (Writing for Computer Science)
  - Don't get too attached to your words. Convey the same idea with more power

References (For This Lecture)

- “Constructing a Good Dissertation”, Erik Hofstee, 2011, EPE Publishers
- “Information Technology Research: A practical guide for Computer Science and Informatics”, Martin S. Olivier, 2nd edition, 2004, Van Schaik
- Slides from Dr Katherine Malan (adapted for this lecture)