The Dynabook as a tool for Research and Development

Introduction

Software is usually seen as the means to solve a problem by computer. Science is usually seen as a buildup of knowledge that can be mined for ideas to solve problems. Keeping Software Development and Scientific Research as two separated activities limits the quality of the former and the applicability of the latter. We describe an approach to scientific research and software development that builds on the ideas and practices of Experimental Mathematics, and best practices in Software Development, and results in an integrated process and tools. The result is a body of knowledge that satisfies the requirements to be considered serious research, and provides useful building blocks for real world software applications.

On Science, Research and Knowledge

Science is a systematic enterprise that builds and organizes knowledge.

Natural Science studies the material world. The knowledge it develops is in the form of testeable explanations and predictions about the universe. A scientific theory is an explanation of an aspect of the world that can be repeatedly tested, using the scientific method. The official body of knowledge of Natural Sciences includes scientific theories, descriptions of the experiments to test them, and reports on the results of those experiments validating a theory, or proving it false.

Formal Science studies abstract systems. The knowledge is in the form of theorems. A Theorem is a statement that can be proved to be correct. The official body of knowledge of Formal Sciences is a collection of such theorems.

But, how is research done? What is the process done by people creating new science?

Research starts with a question. Learning about a field, by looking at the world, studying textbooks or previous research, sets the stage for new questions to appear. Trying to understand something or trying to solve an existing problem will provide new questions.

Once we have some questions, we need to work on them. Sometimes our familiarity with some field and our intuitions are developed enough to just start evaluating possible answers, and filter out many by thinking on them and finding their flaws. Sometimes this is not the case, especially when we are new to some field or some kind of problems. If this happens, the best is to do random experiments. These exploratory experiments are not to validate a theory (like in the Scientific Method) as we still have none. They are just to see what happens. To build our intuitions.

As our familiarity with the field and our intuitions grow, we will start to be able to ask better questions, and to give more possible answers. Further experimentation will allow us to discard some explanations, while giving us more confindence in others. It becomes important for

experiments to be repeatable, and their output to be objectively measured. As we progress, we start building and refining our conceptual models of the behavior we observe. We build knowledge. We *understand*.

Eventually we reach the point where we have an explanation for some phenomena that is consistent with the experiments done, and that is also consistent with already established knowledge, theories and experiments; or that is strong enough to prove them wrong. We have a Scientific Theory.

If we are doing Natural Science, the next step is to document the theory and publish it. This will enable other researchers to reproduce the experiments and confirm the theory or refute it. It will also make it part of the established body of knowledge, part of established science. This will allow further research to be based on it.

But if we are doing Formal Science, what we have so far is a Conjecture and not a Theorem. And as we said before, accepted knowledge in Formal Sciences is theorems and not conjectures. A conjecture, together with supporting evidence in the form of examples and experiments, might be be published, to let other researchers work on it. Still, it will not become accepted knowledge until a formal proof is found. When this happens, the examples, reflections, and auxiliar ideas that helped the discovery of the problem, the solution, and the proof, are removed from the text of the theorem and forgotten, like the scaffolding and formwork used to raise a building.

Digital Image Processing

Here we will focus on open problems in Image Processing, although the methodology can be generalized to other fields. In Image Processing we study the properties of Digital Images and the algorithms that operate on them. The body of knowledge consists not only of the formal properties of images and formally defined operations of them. Images are often captured by real world sensors and cameras, and their properties are related to the physical environment in which images are captured, and the actual (i.e. non ideal) behavior of sensors and optical systems.

In principle, Digital Image Processing is a formal science, a branch of Mathematics. So, it is possible to understand it and develop it in a completely formal way, operating only on what can be dealt with using formal methods, and publishing only polished theorems. But there are several problems with following this approach. The first one is that it leaves out everything that can not be fully formalized, and this means that our work is no longer related to images coming from the real world in a meaningful way. The second problem is that it disregards the very process used to develop it as ultimately useless and not worth sharing. This means that other researchers wanting to work on same or related process must walk the entire way from scratch by themselves. A third problem is that the finished work is something that, while correct, is still a long way from a practical solution to the original problem. This is because the techniques described still need to be implemented on actual computers, and because "on paper, everything works".

Another possible approach is to just build a collection of techniques, in actual code, that seem to solve actual problems in practice. This is a pre-scientific approach, and it is in fact the way

technology was historically developed before modern science. One of the many problems with this approach is that the conditions under which the algorithms are valid are never stated or checked against actual input.

There are groups that follow a more comprehensive and open approach. They publish regular papers, with explanation, formal proofs, and printed example images. But they also publish source code implementing the techniques they describe and the examples used, all available from a web server. A good example is "Interpolation Revisited" by Thévenaz, Blue and Unser. The paper is freely available at http://bigwww.epfl.ch/publications/thevenaz0002.pdf and source code, examples, and instructions to use them are freely available at http://bigwww.epfl.ch/thevenaz/interpolation/ .

Experimental Mathematics and Open Research

Experimental Mathematics is a recent movement in Mathematics that focuses on the process that results on new developments, and not just on the results themselves. It encourages the exploration of conjectures and informal beliefs, using computers to work on many examples. The 'Experimental Mathematics' research journal was once nicknamed 'Journal of Unproved Theorems' by some. The editors answered:

"We are interested not only in theorems and proofs but also in the way in which they have been or can be reached. Note that we do value proofs: experimentally inspired results that can be proved are more desirable than conjectural ones. However, we do publish significant conjectures or explorations in the hope of inspiring other, perhaps better-equipped researchers to carry on the investigation. The objective of Experimental Mathematics is to play a role in the discovery of formal proofs, not to displace them."

The Experimental Mathematics approach states that unfinished ideas and examples are worth documenting and publishing. This reduces the gap between the work actually done while searching for a solution and a proof, and the publication itself. Additionally, the possibility of using a computer both for running experiments and for expressing formal results opens the possibility of integrating them in a common computer model of the field under study, although this is not common practice.

Open Research is a research methodology based on the spirit of Free and Open Source Software. Not only it values the publication of unfinished research: It encourage to distribute ownership of research projects, enabling any individual or group to participate in reseach projects at every stage. We can say that it follows the traditional idea of science as a social construct, but in a much finer grained way. Not only finished projects are published, but every step done.

Approach taken

We take an approach based on the ideas described above. We work on examples, sometimes a massive number of them. We develop computable metrics to assess the properties of our results. We also develop computable mathematical models of sensors, optical systems, and the behavior of light in natural environments. These models are shared with engineers developing cameras and

sensors, and physicists studying the world we take pictures of. The correspondence between these models and reality can not be formally proved, so it is carefully validated through observation.

As described before, we learn by repeating these steps over and over: - Make a question - Take sample images - Attempt tentative answers in the form of algorithms that change and evolve during this process - Make experiments, applying algorithms and taking metrics. During experimentation phase, metrics might be subjective and qualitative, instead of the preferable objective and quantitative. - Do an evaluation of results

It is important to note that the results obtained are only valid for the examples considered, unless we can universally prove them. This doesn't reduce their usefulness. For example, an image registration algorithm can not work if image content doesn't contain enough distinctive features that can be visually aligned. An example might be a set of aerial images of open sea. In these cases, we don't want a formal specification of the input for the algorithm to work. It is preferable for the algorithm to fail gracefully, and handling the failure elsewhere. Future refinements of the algorithm might allow it to succeed in cases where it previously failed.

Doing this, in a documented and repeteable way is enough to build knowledge, to learn, document and publish. Keeping all the stuff on a web server allows following the Experimental Mathematics, Open Research, and building Open Source Libraries. So, what is left to desire?

The Dynabook

An instrument whose music is ideas.

Alan Kay and his team at Xerox PARC developed the idea of the Personal Computer as a new communication media to develop and express ideas. In addition to text, images and conventional media, it will include custom built simulations and digital models. The aim is to be able to model knowledge in software. Not just text, for ideas to exist only inside the mind of the reader, but Computable knowledge. In addition, "authoring is always on". The reader can become author anytime, enhancing or criticizing the content. Books and Essays that are dramatically richer and more powerful than what is possible today.

But the Dynabook is an idea that hasn't been realized yet. Alan Kay says "the computer revolution hasn't happened yet". Besides, most projects inspired by the Dynabook focus on education. So most Dynabook style content in existence is for teaching kids. But the Dynabook, like Vannevar Bush's Memex, is also intended to be used by scientists and engineers.

The closest thing to a general Dynabook that has ever been built is still Smalltalk-80.

Smalltalk-80 enables us to build computable models of any domain under study. Dan Ingalls has said "The system is the curricula", meaning that Smalltalk-80 can be seen as a university level course covering most of Computer Science. In addition, Smalltalk-80 lets us integrate examples, textual descriptions, graphics and complete experiments, even including the automatic validation of the results.

Doing research with a Dynabook

Smalltalk lets us integrate all the research activities in a single environment. We can start with some text, stating objectives, to-do's, questions, etc. We also build a useful collection of sample images. Then we write Test cases to state their properties. We model the required knowledge to work on our problems. For example, we model, document and simulate Cameras (geometry, PSFs), Sensors (geometry, optical and electrical behavior), Real world Scenes (cartography, digital elevation models), Capture geometry in 3D space (camera ECEF position and attitude quaternion), etc. These models hold the algorithms we develop, and allow the simulation of real world behavior of these objects, and also provide services that only can be provided by digital image processing.

So, some of the things we do: - Develop mathematical models of sensors, optics, observed objects and light. Organize them in a coherent framework. - Develop mathematical models of captured images. Implement them in code as simulations. Organize them in a coherent framework. - Develop formal metrics on properties of images. Implement them in code. - Do experiments. These are usually free spirited and don't need to follow structure. - Evolve successful experiments into full fledged algorithms. - Assert the propeties of the results of algorithms. Build a library of experiments and their results, i.e. test cases. Support the application of the Scientific Method. - Formally prove the results of algorithms. - Organize algorithms as part of models. - Organize algorithms in code libraries that are useful for developing applications.

Building Models

To help us understand To express knowledge For expressing experiments For building applications

Documentation of acquired knowledge

Models Dynabook New media

xxxxx

https://en.wikipedia.org/wiki/Experimental_mathematics
https://en.wikipedia.org/wiki/Experimental_Mathematics_(journal)
https://en.wikipedia.org/wiki/Computer_simulation
http://cognitivemedium.com/tat/assets/Kay_What_is_a_Dynabook.pdf
https://en.wikipedia.org/wiki/Scientific_method

Ideas sueltas: - este approach es util tambien para research puro. - este approach es util tambien para desarrollo de aplicaciones puro. - Es una variacion o una aplicacion directa de experimental mathematics? - Es una variacion o una aplicacion directa de agile methods?

Ejemplos. Basico: remuestreo Avanzado: registracion

IR ARMANDO SLIDES

IR BUSCANDO EJEMPLOS EN CUIS #convolutionWith: #simulatePhotonNoise

```
IrregularSampledImage metodos de clase

FloatImage showRotations.
FloatImage showTranslations.
FloatImage showZoomIn.
#displacementFrom: comentario. Aun con ruido, correlacion mejor que 5% de pixel
```