# Construction of a Formal Multimedia Benchmark

Stéphane Marchand-Maillet
*Viper* team - CUI - University of Geneva
CH-1211 Geneva - Switzerland
http://viper.unige.ch/

**ABSTRACT**

The global acceptance of the digital medium to carry information (in particular over the WWW) makes common the use and development of multimedia information processing systems. This class of systems is very wide and numerous instances are presented in the relevant literature. A common scheme for such presentation mostly follows the classical scheme of specification, development and test. However, in most cases, the evaluation of these systems is related to each particular system. Our aim here is to abstract the class of multimedia systems in order to present a base framework via which systems could be formally compared. This paper analyzes the construction of such a multimedia benchmark by looking at the system from different angles and inheriting from techniques in related fields. Examples of known multimedia system benchmarks are discussed.

## 1 Introduction

Nowadays, information is principally carried in digital form. It makes this information both disposable and extremely flexible. First disposable because an unlimited amount of perfect (or similar in some way) copies may be distributed at very low cost. It is also very flexible since it may be carried under several forms and over various supports, the most used being the WWW. We define here multimedia as *any digital storable representation of information.* Multimedia therefore comprises classical images, text, video and audio but may also be richer (eg web pages).

The wide acceptance of multimedia naturally leads to the development various multimedia information management and processing systems. Each of these systems aims at a particular role and is often quite complex. Examples of such systems include:

- Data segmentation and compression systems that process the data at a somehow low level;
- Digital watermarking systems that "stamp" the data with some extra digital information (note that digital watermarking itself may have completely different goals);
- Indexing and retrieval systems that aim at automating (or, at least, easing) the management and search of large multimedia collections.

Multimedia processing systems often arise from the promising perspective of achieving a given task with an expected performance. This task is explicitly (resp. implicitly) defined with respect to an explicitly (resp. implicitly) known form of ground truth. In turn, such a base judgment implies a form of optimal behavior of a multimedia system. While it is generally the case in the literature that the scenario "motivation, development, test, performance analysis" is followed, it is less often clear how to exactly reproduce the results and even less how to compare systems of the same category (ie that are supposed to achieve the same goal). There is therefore a strong need for formalizing the development of evaluation tools that unambiguously allow to characterize the fact that a multimedia system achieves its promises or not.

Benchmarks help evaluating the true performance of systems and hence give insights on the systems' true capabilities, usability and accuracy. When looked at from different perspectives (eg industrial or academic), the benchmarking procedure may have different final aims. While the practical usage of a benchmark may not directly be as such, it typically aims at comparing systems of the same class. In a more practical setup, benchmarks may be used to manage system development by considering a modified system as a "competitor" to the original one (intra-system development). Benchmarks are also useful to better understand the class of systems that is under evaluation. This is partly because the thorough analysis that is presented in this article and is always either explicitly or implicitly done. As a consequence, such an understanding leads to decoupling parts (or aspects) of the system into "orthogonal measurements".

In the sequel, we formalize a generic analysis from which a minimal multimedia system benchmark should be derived. We then detail each component of the benchmark and pinpoint issues. Examples of applications of this formalism are then given and used to extract shortcomings in performance evaluation.

## 2 Abstract structure of the benchmark

Even if the class of multimedia information systems contains differing instances that track a particular aim, we present here a common analysis that should be followed when defining a formal evaluation protocol and platform. We remain guided by the recommendations of J. Gray [1] for characterizing a benchmark (applying on database system) as *relevant* (measuring the right performance), *portable* (system independent), *scaleable* (applying on different system scales) and *simple* (interpreted easily), otherwise it would lack credibility.

We will first characterize the system by looking at the it from various viewpoints or perspectives. For each of these aspects, we wish to express and quantify what is given and what should come as a result. It is then based on this analysis that the benchmark will be defined.

**Operation** Clearly, any given system should aim at satisfying a well-defined goal by performing a targeted operation. Such an operation may be given in a high-level, abstract semantic so that it may be (and generally is) not easy to map it into a set of technical operations. In any case, the definition of the problem tackled by the given class of systems should be well-defined and unambiguously specified. In fact, by fully addressing such an issue, we may loop back and help the design of the system itself (see extended remarks below).

**Data** The data perspective is probably the most important in the design of a multimedia system benchmark. Obtaining a deep and clear understanding of the type of data in question and its flow is crucial for defining principal directions of evaluation. It is in particular from this viewpoint that user input is acquired. Quantifying what should come or be produced automatically and what should come from or be given to the user is important to define the practical setup in which the system should be used.

**System** Still guided by the data flow, one has to define properly and formally the system minimal Input/Output (I/O) configuration. This information flow between the system and external parties (user or benchmark in our case) should be minimally defined so that only useful information is carried in a non-redundant way. Relating also the above issue of operation, there should be a way of considering the system as a "black-box" with such a minimal I/O performing a defined operation, thus leading to a simplistic model.

**Efficiency** Once the system structure and its aims are well-characterized, the goal is now to define the concept of efficiency that will subsequently lead to that of performance. This task is from far the less trivial of the whole process. Modeling efficiency goes down to evaluating the closeness of the system's results to some notion of perfect or optimal performance. The response to this question would lead to the definition of the information space considered and thus would lead to the "closed-form" definition of the perfect targeted system. For example, if the aim of the system is to retrieve images by similarity, an analytical measure of closeness of the system's results to some user-defined ground truth would actually form a distance measure that could itself be used for retrieving visual documents and be a perfect match with the user's definition of perception in this case.

Such a process is clearly unattainable and, there is rather a strong need for defining formal performance measures that lead to unambiguous interpretation of the quality of the system's results. Such performance measures may either be based on comparing the operation done by the system with that done by one or more true users. It can also consist in modeling user judgment in order to assess the intrinsic quality of the results obtained. Clearly, this second solution may be discussable since such performance measures should themselves be evaluated to quantify how well they match with the user's notion of quality. Section 4 refers to some instances of measures (such as precision and recall) that, not only have shown to be efficient as indicators but also can be generalized easily from one topic to another.

**Protocol** Subsequently to the definition of the benchmarking procedure, one should define the benchmark protocol (eg [7]). At least two aspects should be considered. One should look at parts of the benchmark that could be automated so as to easily rerun the evaluation process or reproduce results. Another aspect is security. Whenever the benchmarking protocol is automated, one should make sure that this evaluation cannot be rerouted, if this is important in the given context. However, one should always consider that there are little incentives in defeating benchmarking systems since this will always be discovered, corrected and counter-publicize (if that was the aim).

Based on the above analysis, Figure 1 gives the abstract structure of a general benchmarking system. Typically, a central *knowledge repository* should store the base knowledge about the *system expectation* and configuration. This knowledge should be related to *standard data* on which *standard tests* will be based. By providing the tested system with this data, we get the resulting output that should be evaluated with referring back to the base knowledge. This structure is a base for a formal

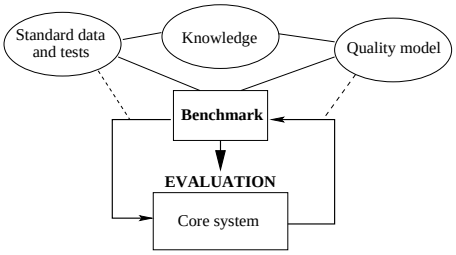


Figure 1: Multimedia benchmark design.

architecture when practically addressing benchmarking

issues.

**Related remarks** Multimedia systems are complex systems integrating various components and should be treated as such. The field of software engineering commonly deals with the management of such complex systems where various information flows can be characterized. Formalisms like UML have helped a great deal in coping with the complex and diverse aspects of software management. Like in our case, software developers have faced the problem of measuring unambiguously the performance (or rather conformance) of their systems with some form of ground truth (ie specifications in this case). A very interesting approach has led to the development of the "test first design" with the XProgramming community.[1] Such an approach consists in designing a testbed *prior to* developping each component of the system. These test sets are referred to as *unit tests*. In other words, the approach is to concentrate on *what* should be achieved rather than *how* it should be achieved. When following this formalism, at the end of each development cycle, the set of tests associated with each component could be called a form of benchmark for the system in question.

What is more interesting is the fact that this approach triggered the development of the so-called XUnit context, which aims at developing a set of tools for easy the creation of unit tests. Multimedia system developers could therefore inherit from two main aspect of this approach. Firstly, by designing tests first (ie characterizing what should be achieved), this leads to a better comprehension of all aspects of the system and will therefore enhance its development. Further, the development of a standard set of unitary tools to evaluate basic components would lead to the easy generation of multimedia system benchmarks.

## 3 Addressing issues

We now analyse issues that are related or triggered by the benchmarking context.

**Knowledge input** Multimedia systems are aimed at performing a given operation. This operation should suit users' needs. Hence, user needs should be described somewhere in the system. Two typical ways of representing user's knowledge are generally used, namely using direct user assessment or designing performance measures. Having unambiguous performance measures combined with standard test operations is crucial in the design of a benchmark. Whenever main characteristics of the system are derived, they should be mapped into one or more such measures so that a *set of measurements* is used to qualify the system. A *lead* measure may also be used are main indicator for system comparison.

**Standardize I/O** As defined above, a multimedia system is a process that accepts some input and produces a given type of output. Any communication with the system is therefore done based on these two input and output streams. One key for the success of a system benchmark is its simplicity within the setup. By analyzing the class of multimedia system in question and proposing a standard form of communication, one facilitates the subsequent "connection" to a (possibly automated) benchmark. The XML technology offers now a number of advantages for defining standard ways of exchanging data. Systems where communications are defined using such technology therefore facilitate greatly the construction of benchmarks. It is on the other hand to the benchmark designers to recommend standard I/O for the class of systems in question to promote compatibility. This is the case in the examples presented below where languages like MRML[6][2] are used. Other SOAP and, more generally, XML protocol languages give insights on how to standardize the access to a system.

**Publicize it** Gaining public acceptance is one challenging task. Having the benchmark developed in a wide open collaborative environment surely helps the process. However, this moves the burden to that of motivating teams to participate into the development of the platform.

The issue of attracting participation goes beyond simple communication. It includes the definition of the protocol for using the benchmark and displaying the results. Whether results should be made public or not is one issue. Also graphical result representations should be considered carefully. This aspect typically helps a great deal in performance interpretation and thus makes it possible to pinpoint un/efficient techniques.

Figure 2 gathers the above analysis into a schematic multimedia system benchmark. All aspects (materialized by components) of this architecture should be addressed carefully in order to obtained maximum flexibility and relevance of the benchmark, thus matching with our criteria for a good benchmark [1].

## 4 Examples of benchmarks

We list here relevant examples of benchmarks that are under development or already successfully applied. These examples are linked with their respective organization, which generally have a non-profit policy. We refer the reader to their respective main public sites to get details on their architecture. Typically, under a form or another all these structures implicitly or explicitly match the above analysis.

**Benchathlon** The Benchathlon network[3] is a joint international effort to create a standard platform for the evaluation of Content-based Image Retrieval (CBIR) systems. It is still at a construction stage so that it concentrates on gathering contributions on the benchmarking methodology. However, a freely available image

[1] http://www.xprogramming.com

[2] `http://www.mrml.net`

[3] `http://www.benchathlon.net`

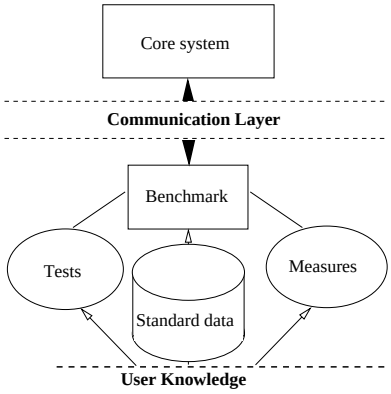


Figure 2: Multimedia benchmark formal architecture.

collection is gathering and a context for annotating it is under setup.

The protocol and setup used for the Benchathlon corresponds very much to that presented in [4] (conform to our analysis), although a first proposal was given in [2]. The principle is to consider the CBIR system as a web-accessible query processor and the benchmark as a web-based client. Both are connected via a standard communication protocol (MRML) and a fully-automated evaluation procedure is started. In this setup, CBIR system designers should ensure that their architecture is compatible to the automated framework presented.

Performance measures proposed are mostly based on the notion of *precision* and *recall* that are relevant in the information retrieval field [2, 3, 5]. These measures are used jointly and read in precision-recall (PR-) graphs. Other indicators aim at accounting for system usability and should all be read in conjunction of one to another.

**TREC** A related wider initiative is that of TREC[4]. It consists in a series of conferences where information retrieval systems are evaluated. TREC started with text retrieval (TR) systems and has extended to web-pages and video, via so-called tracks (no image track is yet defined).

TREC has simplified the evaluation procedure, it proposes no automated evaluation but an elaborated sequence of user-based assessment. TREC concentrates in providing developers with annotated test data and performance measures so that only results are submitted and presented in an informal contest at the TREC event. TREC also makes extensive use of *precision* and *recall* measures.

**SPECMedia** The aim of SPEC[5] is to "*establish, maintain, and endorse a standardized set of relevant benchmarks and metrics for performance evaluation of modern computer systems*". It mostly concentrates on the evaluation of commercial (hardware) systems (as opposed to academics software prototypes) and proposes a number of benchmarks to evaluate components ranging from CPUs to displays. More specifically, SPECMedia is sub-initiative to benchmark multimedia systems. Its first target concentrates on MPEG2 video with generic attributes (MP@ML). Although promising, little is given on the progress of this development.

Other benchmarks exist in various domains such as image watermarking[6] or database transaction[7] .

## 5 Conclusion

Multimedia information processing systems are now commonplace. For each processing task, various (often very different) alternatives strategies are presented. This makes it difficult to compare their respective performance and, more importantly to assess their advantages and shortcomings. In this paper, we formalize a minimal procedure via which a multimedia system benchmark should be constructed and used. Examples of existing evaluation procedure show that this formalism is actually already implicitly or explicitly used and has lead to successful setups. The generalization of such a procedure is, in our view, the key for successful efficient, useful and usable multimedia information processing applications.

## References

[1] J. Gray and R. Cattell. *The Benchmark Handbook.* Morgan Kaupfman, 2nd edition, 1993.

[2] N. J. Gunther and G. Beretta. A benchmark for image retrieval using distributed systems over the internet: BIRDS-I. Technical report, HP Labs, Palo Alto, Technical Report HPL-2000-162, San Jose, 2001.

[3] C. Leung and H. Ip. Benchmarking for content-based visual information search. In Robert Laurini, editor, *Int. Conf. On Visual Information Systems (VISUAL'2000)*, (LNCS 1929), pages 442–456, Lyon, France, 2000.

[4] H. Müller, W. Müller, S. Marchand-Maillet, D. Squire, and T. Pun. A web-based evaluation system for content-based image retrieval. In *ACM Multimedia Information Retrieval (ACM MIR 2001)*, pages 50–54, Ottawa, Canada, 2001.

[5] H. Müller, W. Müller, D. Squire, S. Marchand-Maillet, and T. Pun. Performance evaluation in content-based image retrieval: Overview and proposals. *Pattern Recognition Letters*, 2000.

[6] W. Müller, Z. Pečenović, H. Müller, S. Marchand-Maillet, T. Pun, D. Squire, A. P. De Vries, and C. Giess. MRML: An extensible communication protocol for interoperability and benchmarking of multimedia information retrieval systems. In *SPIE Photonics East - Voice, Video, and Data Communications*, pages 961–968, Boston, USA, 2000.

[7] R. Ruiloba, P. Joly, S. Marchand-Maillet, and G. Quenot. Towards a standard protocol for the evaluation of temporal video segmentation algorithms. In *European Workshop on Content-Based Multimedia Indexing, CBMI'99*, Toulouse, France, 1999.

[4] `http://trec.nist.gov`
[5] `http://www.spec.org`
[6] `http://www.certimark.org`
[7] `http://www.tpc.org`