

Distributed Intelligent Medical Assistant for Osteoporosis Detection

Wided Lejouad Chaari, Sabine Moisan, Sylvie Sevestre Ghalila, and Jean-Paul Rigault

Abstract—This paper describes a distributed knowledge-based system managing medical image processing programs to assist physicians in establishing image analysis protocols. We rely on program supervision techniques that aim to automatically plan and control complex software usage. Our main contribution is to allow physicians who are not experts in computing to efficiently use various osteoporosis detection programs in a distributed environment. A distributed supervision system allows either to simply consult knowledge bases, or to execute remote queries, or to collaborate with distant partners to design knowledge bases and programs.

I. INTRODUCTION

IN the domain of osteoporosis, diagnosis is of major importance. The most challenging task is to characterize bone micro architecture by parameters that can be automatically estimated from images and that can accurately detect and quantify alterations of bones. For this, we have developed an original approach using morphological tools to extract characteristic features of trabecular bone images [1]. To make such an approach operational for medical diagnosis, it is necessary to determine an “image protocol” adapted to bone types (e.g. femur, wrist, vertebrae) and patient types (e.g. male or female, adult or child) for various image resolutions. Setting such an image protocol consists in planning a sequence of programs and tuning their input values (e.g., threshold values, filter size, etc.). This constitutes a tedious, time consuming task which requires both clinicians and image processing experts to collaborate. Our solution is to provide an interactive tool which relies on artificial intelligence techniques to build image protocols in different situations.

Moreover, we wish to make this system accessible to bone radiologists, a geographically scattered community. Hence, this paper proposes a *distributed intelligent assistant* for osteoporosis detection, relying on mobile agent technology. This assistant provides a collaborative framework driven by the following desirable properties:

- To assist clinicians in image protocol management, freeing them from performing computer science tasks, yet giving them access to up to date image processing.
- To allow experts in medical imaging to compare algorithms and to improve them incrementally, thanks to feedbacks from clinicians.
- To allow knowledge engineers to collaboratively construct knowledge bases for medical imaging.
- To do all the above, with people and resources distributed all over the world.

In the following, section II briefly describes the image processing algorithms, and section III defines program supervision. The next section (IV) details the architecture of the assistant and the knowledge that it embeds. Section V addresses the problems related to distributing the system and shows a scenario of use.

II. MEDICAL IMAGE ANALYSIS TO DETECT OSTEOPOROSIS

Many computer-aided methods of X-ray micro-architecture bone analysis have been developed to complement the Bone Mineral Density measurement technique. We can divide them into two classes: texture analysis and morphological analysis. Our osteoporosis detection process belongs to the latter class. It consists in extracting morphometric features from a grey or binary skeleton. To this end, we propose several image processing programs summarized on figure 4. These programs often depend on tunable input parameters and some of them correspond to alternative algorithms for the same functionality. Setting a protocol thus requires tuning the parameters and choosing among the alternatives. To illustrate it we give a brief description of two important processing parts, namely, *skeletonization* and *analysis* of the resulting skeleton.

Skeletonization consists in thinning iteratively the image foreground (the trabeculae bone). The foreground is obtained by a thresholding step [1]: we compare the original image (in this case a calcaneum image, figure 1(a)) to its local p -percentile computed on a sliding window of size w . Then two thinning methods are proposed: a classical binary skeletonization and a grey thinning close to the Mersal algorithm [2]. The latter consists in two steps. First, the weakly connected pixels at the interface between foreground and background (“border pixels”) are removed and their grey values are stored and ranked. Second, scanning these stored values with a step s , other weakly connected border pixels can be removed. The step s corresponds to the expected thinning precision.

The analysis process starts by splitting the skeleton into

Manuscript received April 16, 2007. This work was supported by the cooperation project STIC 03/106 between INRIA-France and ENSI-Tunisia.

W. L. Chaari is with the National School for Computer Studies, SOIE Laboratory, Manouba University, 2010 TUNISIA (phone: 216-71-600444; fax: 216-71-600449; e-mail: wided.chaari@ensi.rnu.tn; chaari.ite@planet.tn).

S. Moisan is with National Research Institute in Computer Engineering and Control (INRIA), ORION Project, Sophia Antipolis 06902 FRANCE (e-mail: sabine.moisan@sophia.inria.fr).

S. S. Ghalila is with National School for Engineers, U2S Research team, Tunis El Manar University, 1002 TUNISIA, MAP5 Lab. Paris 5 University, 75270 FRANCE (e-mail: sylvie.sevestre@math-info.univ-paris5.fr).

J. P. Rigault is with Nice University, FRANCE (e-mail: jpr@essi.fr).

sets of pixels: the set of intersection points and the segments that connect such points. The result is illustrated in figure 1(b). The segments are labeled according to their orientation as compression (vertical segments, related to the patient's weight) or tension (horizontal, due to walking force). At this stage, morphological features providing reliable information about bone micro-architecture are computed for each segment. An inter-projected distance is estimated as the length of a segment. An intra-projected distance is also computed by measuring the distance between the two lines of maximum gradient directions around the segment. The gradient reliability is controlled by two parameters ϵ_{ns} for the tolerance of the gradient line position and ϵ_{nl} for the tolerance of the normal to the gradient line used to increase the reliability of the maximum gradient detection.

This whole analysis sequence corresponds to the *DimProj* program indicated in figure 4. The other programs on the figure also require the same sort of decisions (parameter tuning, algorithm choice) to set the image protocol. This is the role of the intelligent assistant described in section IV.

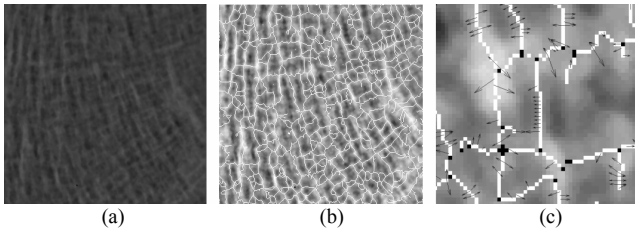


Fig. 1. Image (b) is the grey skeleton extracted from the original image (a). Image (c) is a zoom of a part of (b) where the intra-projected distances are represented by arrows.

III. PROGRAM SUPERVISION SYSTEMS

A. Definition

Program supervision is an artificial intelligence approach to automatically select, monitor and execute programs that process data. This is done by a knowledge-based system that emulates the strategy of an expert using the programs, i.e. selection, combination, execution, and control of programs. In this approach, programs are considered as “black boxes” and their application domain or their programming language is not relevant. The goal is not to optimise the programs themselves, but to assist program *usage* [4].

Indeed, in non-trivial applications, as medical imaging, several programs have to be scheduled and their execution monitored, in order to perform the best possible processing. Moreover, individual programs may be difficult to handle by non-specialists, because of sensitive parameters tuning or complex applicability requirements. A knowledge-based approach can embody this expertise to produce a system integrating the programs and their use.

Such systems help establish “image protocols”, in the sense that they produce plans of programs with tuned input parameters. Thanks to expert knowledge, this is done automatically, thus not requiring users to be familiar with technical details of the algorithms.

B. Knowledge for Program Supervision

A Program Supervision Knowledge-based System (KBS) is composed of an inference engine operating on a knowledge base. The base formalizes the necessary knowledge about programs and their use, mainly,

- primitive operators that are individual descriptions of programs including their input/output arguments, the types and (default) values of parameters, the syntax to run the code, etc.;
- composite operators that are skeletal plans of operators (sequences, alternatives, etc. of programs) and data flows among these operators;
- decision rules describing how to choose among alternatives, how to adapt parameter values, how to assess a result, or how to repair a problem.

This information is given only once, by experts in the program use (experienced users or program designers), independently of any data. The only information that end-users must provide is the data of the particular cases to run.

Program supervision may be applied to different domains, such as image or signal processing, or scientific computing. For our part, we have worked on program supervision for more than 20 years. Currently, we offer the Pegase+ supervision engine which has been used in several applications, namely, road obstacle detection, medical imaging [6], and astronomy imaging.

IV. OSTEOPOROSIS DETECTION ASSISTANT

Our system for supervising osteoporosis programs, as illustrated in figure 2, includes a knowledge-based system, the osteoporosis programs described before, and a graphic interface. The KBS itself is composed of a knowledge base run by the Pegase+ inference engine that implements the program supervision task.

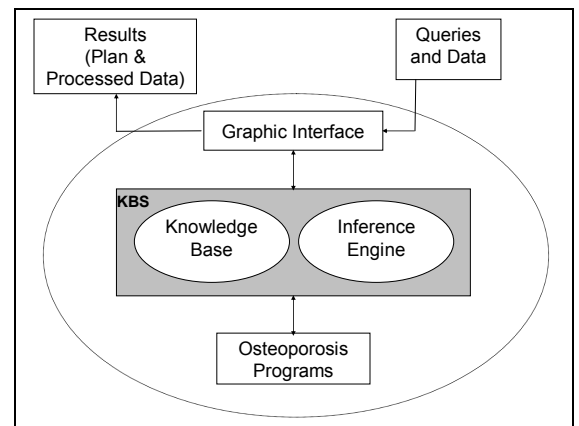


Fig. 2. Supervision system assistant

The contents of the knowledge base allows the KBS to choose, apply and adapt the osteoporosis programs to process numerical radiographic images provided by end users. This work gathers artificial intelligence and medical image processing experts collaborating with physicians specialized in osteoporosis diagnosis; thus, the current knowledge base incorporates the expertise from our image

processing and medical partners [1].

The knowledge base is developed using a dedicated knowledge representation language, named Yaki [5], which uses both object-based and rule-oriented descriptions. This language allows experts to define domain types, objects or operators, and different rules to be used during reasoning. For instance, here follows an example from the osteoporosis knowledge base: a composite operator that describes an alternative decomposition (denoted by a \mid) into two sub-operators: *grey-level* or *binary skeletonization*.

```

Composite Operator { name skeletonization
comment "skeletonization of grey image"
Input Data
  Symbol name grey_image
Output Data
  Symbol name skeleton_image
  Integer name nb_connect_parts
Body GreySkel  $\mid$  BinSkel
Choice Rule { name greyChoice
  If ask_user "Do you need to compute the number
    of connected parts ?" ["yes" "no"] == "yes"
  Then use_operator GreySkel }
Data Flows
  skeletonization.image_grey / GreySkel.grey_image
  skeletonization.image_grey / BinSkel.grey_image
  ... }

```

A choice rule has been given by the expert to decide which sub-operator should be selected depending on the situation (in this case the choice is left to the end-user). The *Data Flows* part displays information about data transfer between the parent operator and its sub-operators.

The role of rules is essential to the engine strategy at different points during the reasoning process: for instance, to choose between several alternatives (choice rules, as shown above), to adapt program execution, to assess result quality, and to repair a badly assessed execution. These decision points are the key to establishing and adapting the “image protocols”.

V. DISTRIBUTED MEDICAL ASSISTANT

In our previous applications we used a centralized supervision system in which data, programs and KBS have the same location. However, applying program supervision to osteoporosis detection requires distributing the system. Indeed, both data (images from different hospitals) and programs (developed by different teams) come from various places. In most cases it would be fairly inefficient to move data and executable code to the same place as the KBS, and in some case it is even not possible (e.g., programs may execute only on a specific hardware). Therefore, we have developed a distributed version of the assistant, based on mobile agents, where clinicians from different countries may safely use the system, medical imaging experts may manage different versions of their programs, and knowledge engineers may capitalize knowledge and adapt it to program changes.

A. Architecture of the multi-agent system

We first developed a Web server allowing remote users to access the supervision system. Second, we distributed the components of the supervision system itself, implementing

the communications with mobile agents [3]. Meta-data help localize the various resources (programs, data, execution servers...).

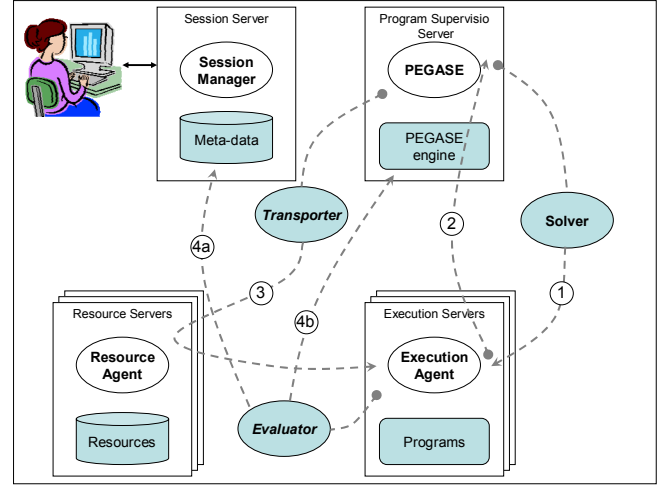


Fig. 3. Architecture of the multi-agent system and scenario of use

The architecture of the system as presented on figure 3 consists of several types of servers. First, one *Program Supervision server* runs the Pegase+ engine; then possibly several *Execution servers* enable to execute the planned programs; *Resource servers* contain remote resources needed by executions (e.g., data files, scripts); finally, the end-user interacts directly with a *Session server*.

B. Agent roles and communications

A *Session Manager* agent is in charge of the whole user session. It is deployed first and initializes the whole system: it localizes the various servers and dispatches other agents. Then the Session Manager waits for a user request and will be responsible for controlling the request processing.

The Session Manager creates a *Solver* (mobile) agent for each request which goes back and forth from the Program Supervision server to some execution server (paths 1 and 2 on figure 3). It obtains the next step of the supervision plan from the Pegase+ engine, and it goes to the execution servers to execute the requested programs, transporting intermediate results and execution parameters. A computation may require remote resources: it is the role of the *Transporter* agents to fetch them (path 3).

When computation ends, according to expert rules, the Solver may create an *Evaluator* agent to assess the result. The Evaluator communicates the assessment either to the supervision engine or directly to the user. In case of unsatisfying result, a repair action can then be performed, either manually (path 4a) by means of end user interaction, or automatically by the supervision engine (path 4b).

C. Medical Scenario

This section illustrates a rheumatologist processing an image through our distributed medical assistant. After connecting to the supervision server, the rheumatologist simply uploads an image to analyze.

Then the knowledge-based system launches the first

composite operator, *OsteoMorph*. The system then starts a planning phase by decomposing this operator into its sub-operators, as shown in Figure 4: *Reading*, *Skeletonization* and *Analysis*. *Reading*, (1) in figure 4, is a primitive operator and does not need evaluation. The second sub-operator (*Skeletonization*) is a composite one, the system has to choose between two alternative sub-operators: *BinSkel* or *GreySkel* (i.e. binary or grey-level skeletonization). Using expert choice rules given in the knowledge base, the engine selects one of them, say *BinSkel* (2). Since the knowledge base requires an evaluation of result for this operator, planning must be suspended and execution performed.

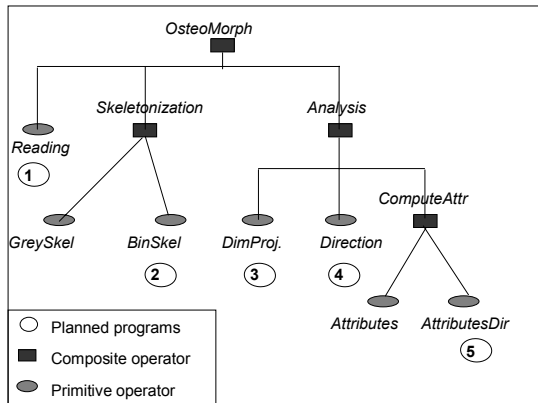


Fig. 4. The *OsteoMorph* operator decomposition.

Execution starts with the Solver agent moving to the *Reading* program site and running it. Then the Solver goes to the *BinSkel* program site to execute it. Now, the evaluation phase can be performed. An Evaluator agent is created to execute evaluation rules. In case of automatic evaluation, it moves to the engine site to execute them. In case of user evaluation, it moves to the server site in order to ask the questions provided by these rules to the user. Then the Evaluator sends the assessment back to the engine. If the assessment is good, the planning phase continues with the *Analysis* composite operator. Otherwise, repair or adjustment rules are used to decide to modify the plan or to re-execute the same programs with different parameter values, resulting in a new plan. For instance, *BinSkel* assessment is manual. The user is asked about the presence of undesirable segments. If this is the case an adjustment rule is fired, that increases the value of a pruning parameter, so that *BinSkel* may be executed again.

The same process runs up to the last program in the decomposition (5). The Solver agent executes this last operator and moves back to the server with the final result together with the plan established by the engine, for example, the following sequence of programs: *Reading*, *BinSkel*, *DimProj*, *Direction* and *AttributesDir*.

The system may also have other uses depending on the end-user type. For instance, image processing experts can use the supervision server to check their programs, observe their results, and compare them with other approaches. A collaborative construction of knowledge bases on the use of medical image processing programs is also possible.

VI. CONCLUSION

Early osteoporosis detection is a challenge for the health community. In this paper, we have described an intelligent interactive system relying on knowledge based techniques: (1) to assist physicians and image experts in validating “image protocols”; (2) to facilitate access to up to date image programs by rheumatologists who do not want to bother about medical image analysis details; (3) to allow physicians to submit an image, to obtain resulting medical parameters and also the corresponding execution plan (that is the effective “image protocol”).

At this time, the system has been tested internally using some real radiographic images. In the near future, it will be made available on the Web to physicians so that they may process their own images and contribute to its evaluation.

Distributing such systems is fundamental because physicians, image processing programs, images, inference engine, knowledge bases, etc. are generally located at different sites. Our distributed environment is based on a Web server and mobile agents. We are currently studying security and confidentiality issues. In the near future, we also plan to use Semantic Web technology to facilitate physician access and knowledge exploration.

ACKNOWLEDGMENT

We kindly thank Pr. C. L. Benhamou and Dr C. Chappard (CHR Orléans, France) for their interest in our work and for providing radiographic images.

REFERENCES

- [1] S. Sevestre-Ghalila, A. Benazza, A. Ricordeau, N. Mellouli, C. Chappard and C. L. Benhamou, Texture image analysis for osteoporosis detection with morphological tools. MICCAI 2004, volume 1, pages 87-94, LNCS 3216, Springer Verlag, Sep. 2004.
- [2] S. S. Mersal and A. M. Darwish, “A new parallel thinning algorithm for gray scale images,” *IEEE Nonlinear Signal and Image Proc. Conf.*, Antalya, Turkey, June 1999.
- [3] N. Khayati, W. Lejouad-Chaari, S. Moisan, J. P. Rigault. “Distributed Knowledge-based Systems Using Mobile Agents”. WSEAS Trans. Computers, Vol. 5, N° 1, pp 22-29, Jan. 2005.
- [4] M. Thonnat and S. Moisan, “What can Program Supervision do for Software Reuse?,” *IEEE Proc. Special Issue on Knowledge Modelling for Software Components Reuse*, Vol. 5, No. 147, pp.179-185, Oct. 2000.
- [5] S. Moisan, “Knowledge Representation for Program Reuse”, ECAI’02, Lyon France, Jul. 2002.
- [6] M. Thonnat, S. Moisan and M. Crubézy, “Experience in Integrating Image Processing Programs”, Int. Conf. Vision Systems (ICVS’99). Las Palmas, Spain. January 1999. LNCS 1542, pp 200-215.