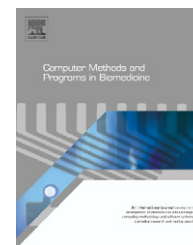




ELSEVIER

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

Multimod Data Manager: A tool for data fusion

Marco Viceconti^{a,*}, Fulvia Taddei^a, Laura Montanari^a, Debora Testi^a,
Alberto Leardini^b, Gordon Clapworthy^c, Serge Van Sint Jan^d

^a Laboratorio di Tecnologia Medica, Istituti Ortopedici Rizzoli, Via di Barbiano 1/10, 40136 Bologna, Italy

^b Laboratorio di Analisi del Movimento, Istituti Ortopedici Rizzoli, Italy

^c Department of Computing and Information Systems, University of Bedfordshire, United Kingdom

^d Department of Anatomy, Université Libre de Bruxelles, Belgium

ARTICLE INFO

Article history:

Received 7 February 2006

Received in revised form

10 May 2007

Accepted 10 May 2007

Keywords:

Data fusion

Computer aided medicine

Musculoskeletal apparatus

ABSTRACT

Nowadays biomedical engineers regularly have to combine data from multiple medical imaging modalities, biomedical measurements and computer simulations and this can demand the knowledge of many specialised software tools. Acquiring this knowledge to the depth necessary to perform the various tasks can require considerable time and thus divert the researcher from addressing the actual biomedical problems. The aim of the present study is to describe a new application called the Multimod Data Manager, distributed as a freeware, which provides the end user with a fully integrated environment for the fusion and manipulation of all biomedical data. The Multimod Data Manager is generated using a software application framework, called the Multimod Application Framework, which is specifically designed to support the rapid development of computer aided medicine applications. To understand the general logic of the Data Manager, we first introduce the framework from which it is derived. We then illustrate its use by an example—the development of a complete subject-specific musculo-skeletal model of the lower limb from the Visible Human medical imaging data to be used for predicting the stresses in the skeleton during gait. While the Data Manager is clearly still only at the prototype stage, we believe that it is already capable of being used to solve a large number of problems common to many biomedical engineering activities.

© 2007 Elsevier Ireland Ltd. All rights reserved.

1. Introduction

The work of the biomedical engineer is increasingly based on the simultaneous use of data sets that have come from disparate sources. It is common to use data from multiple medical imaging modalities, and to combine these with biomedical measurements and results from computer simulations. This poses a number of technical problems including the interpretation of different storage formats, spatial and temporal registration, feature extraction, classification and segmentation, interactive visualisation, etc.

There are specialised tools that may help to solve each of the individual problems, but this implies that we routinely have to use a large number of software tools that are, in general, expensive, demanding to learn, and difficult to combine into a smooth or semi-automatic workflow.

Examples of the various software categories are commonly referred to in the literature. AVS/Express [1], OpenDX [2], and SCIRun [3] are general-purpose scientific visualisation environments organised around the data flow paradigm. Amira [4], 3D Doctor [5], and Mimics [6] are 3D visualisation applications that excel in image processing activities such as segmentation.

* Corresponding author. Tel.: +39 051 6366864; fax: +39 051 6366863.

E-mail address: Viceconti@tecno.ior.it (M. Viceconti).

0169-2607/\$ – see front matter © 2007 Elsevier Ireland Ltd. All rights reserved.

doi:10.1016/j.cmpb.2007.05.002

3D Slicer [7] is an application mostly focused on neurosurgery problems. Other programs, such as OsiriX [8] are specialised in the visualisation of medical images only. There are some solutions that provide an integrated environment, e.g. Analyze [9], but they are limited to medical and geometric imaging. A promising environment still under development, the Julius framework [10], aims to provide a more integrative approach but it mainly focuses on medical imaging and computer-aided surgery.

The present study describes a new application, the Multimod Data Manager (hereafter simply called the Data Manager), which is distributed as freeware. The Data Manager provides the end user with a fully integrated environment for the fusion and manipulation of all biomedical data and does not necessarily demand any knowledge of programming from the user.

2. Description of the Data Manager

The Data Manager is an application developed using a software application framework called the Multimod Application Framework (MAF), which is specifically aimed at providing a supportive environment for the rapid development of computer aided medicine applications. A detailed description of MAF and its capabilities is presented in [11]. MAF is currently being developed through the OpenMAF Open Source collaborative development initiative [12], and distributed under a BSD-like license, which allows the use of MAF for the royalty-free development of commercial applications.

Although MAF is a cross-platform software environment that can be compiled under multiple operative systems, the Data Manager is currently developed only for Microsoft Windows operating systems. All examples in this paper were produced using a standard personal computer equipped with an Intel P5 processor running at 2.8 GHz, 1 GB of RAM, a 120 GB hard disk, and a Radeon 9200 Pro graphic board, running at a screen resolution of 1024 × 768. We used Data Manager v1.2.1 running under Microsoft Windows XP SP2.

At launch, the Data Manager exposes a relatively simple user interface (Fig. 1). A menu bar is placed at the standard position in the upper part of the screen, while a message area is positioned at the foot. On the right of the screen, there are four tabbed frames, named Tree, View, operation and VME. These correspond to the four fundamental concepts that one needs to master to use the Data Manager effectively. The remainder of the screen is left free for the various Views that the user can open while using the program.

An essential feature of the Data Manager is its ability to import virtually any type of biomedical data into a hierarchical data structure in which each data block is called a Virtual Medical Entity (VME). Each VME contains a dataset, a pose matrix that defines the position and orientation of the dataset, and a number of metadata attributes (textual information associated with the data themselves). Both the pose matrix and the dataset can be time varying.

The Data Manager can import 3D volumes generated from almost any medical imaging modality (CT, MRI, PET, SPECT, 3D Ultrasound) and written in DICOM format, the interna-

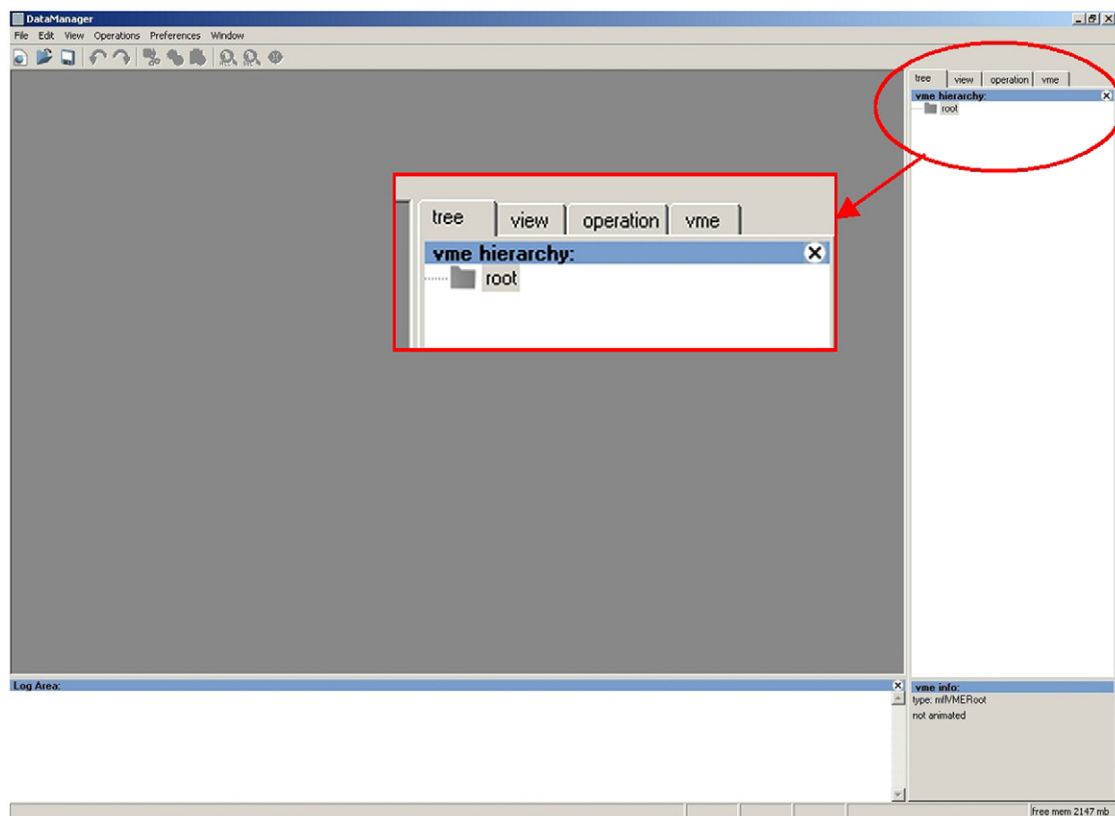


Fig. 1 – The Data Manager launch interface; the four tabbed frames are shown enlarged.

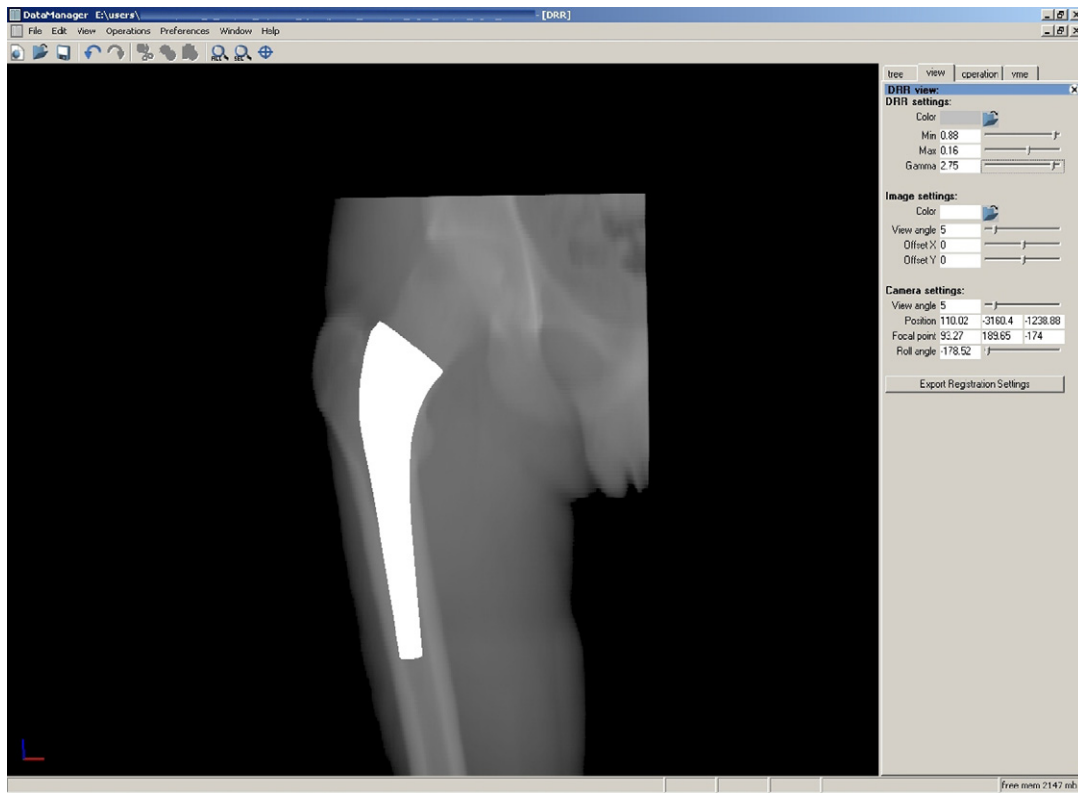


Fig. 2 – Digitally Reconstructed Radiograph View of a CT scan volume and the polygonal surface model of a surgical implant; the parameter-setting panel of the DRR view is visible on the right.

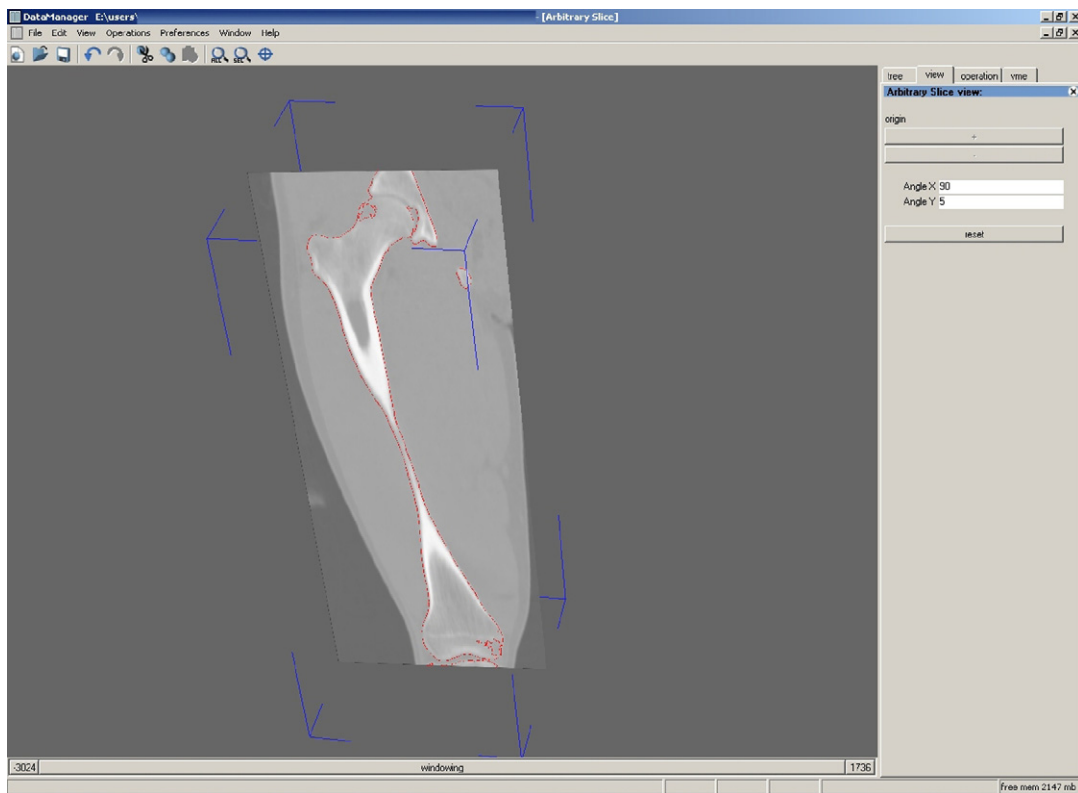


Fig. 3 – The Arbitrary Slice View in which the CT scan volume is cut, together with the related isosurface (shown in red) by an arbitrary plane defined by the user; the panel to adjust the cutting plane is visible. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of the article.)

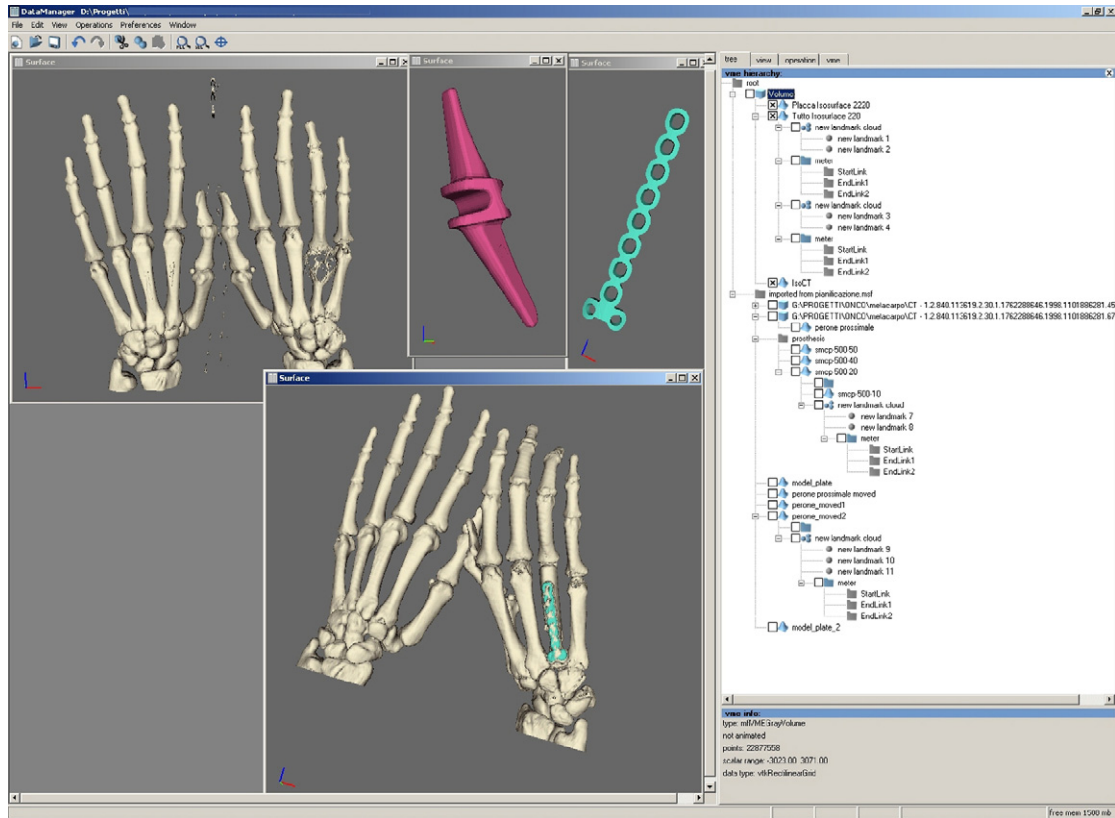


Fig. 4 – Different VMEs are shown using multiple instances of the Surface View.

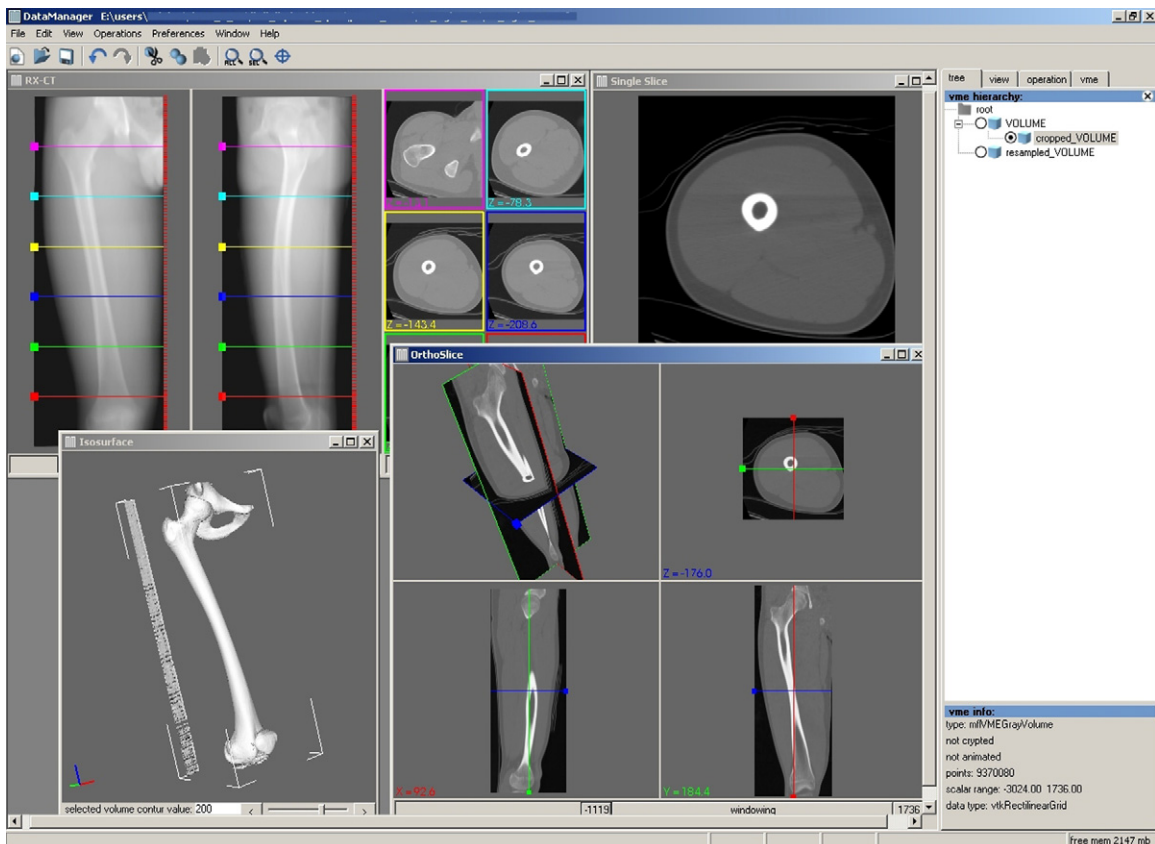


Fig. 5 – The same VME (a CT scan volume) is shown using different Views; clock-wise from top left: an RX-CT view, a Single Slice view, an OrthoSlice view and an Isosurface view.

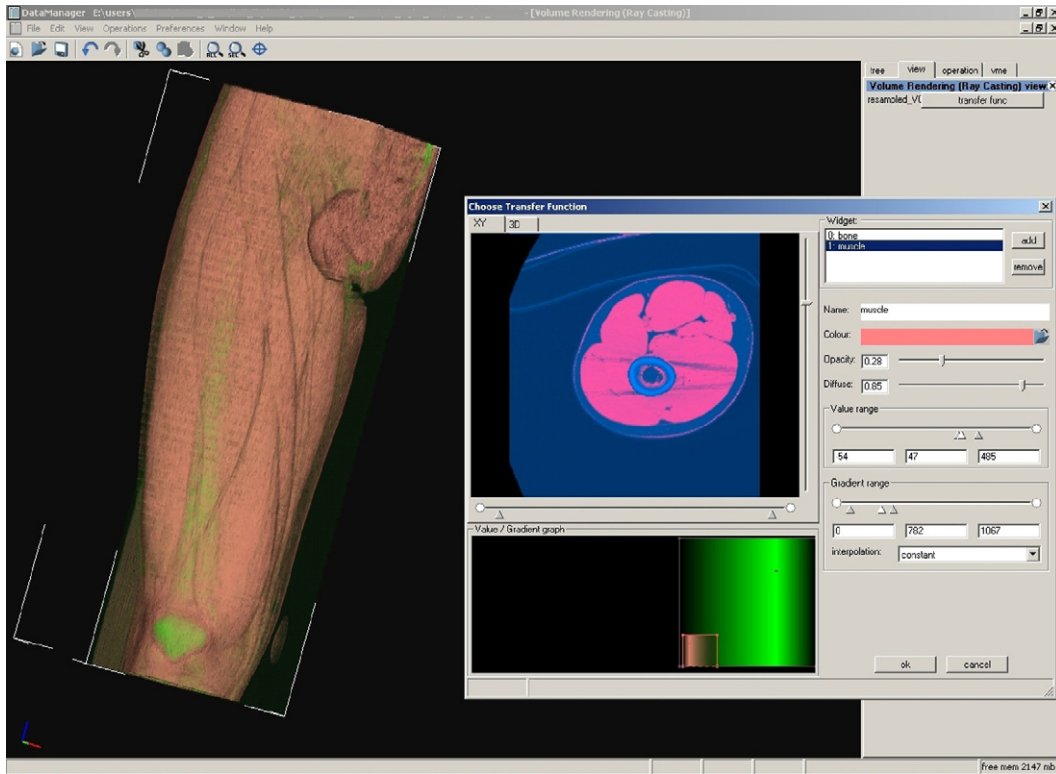


Fig. 6 – A CT scan volume visualised with the Volume Rendering View; the transfer-function panel is visible on the right—different transfer functions were used to visualise muscles (red) and bone (green). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of the article.)

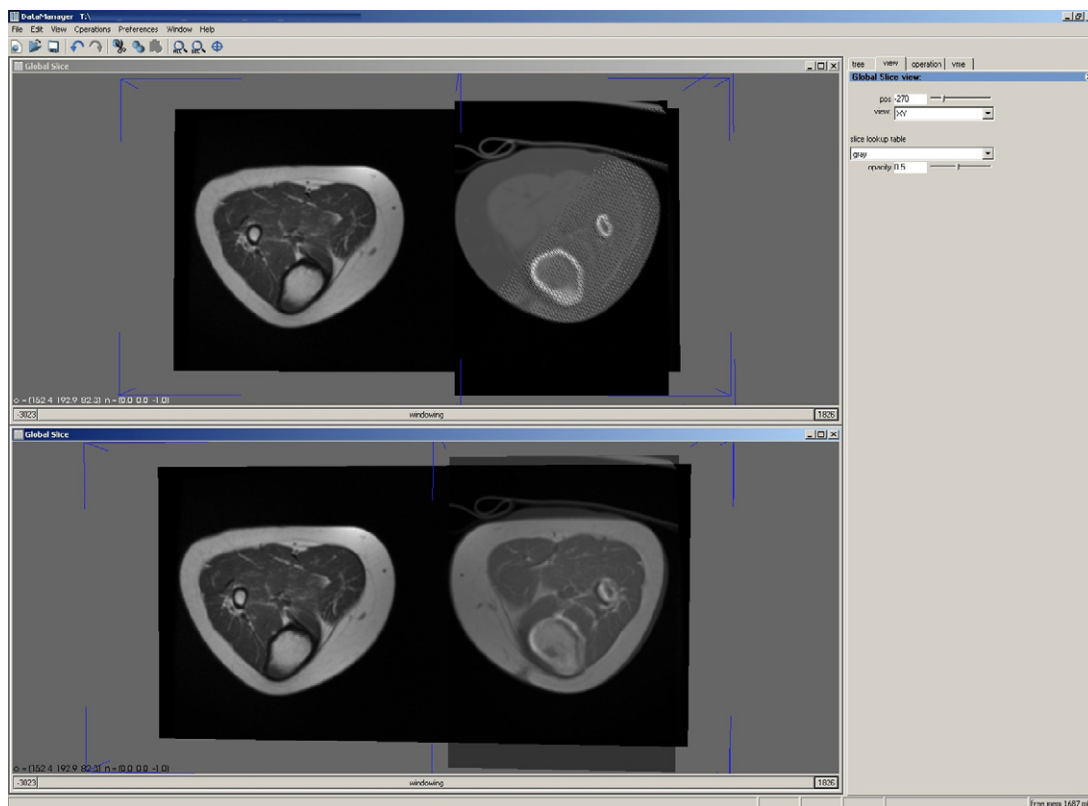


Fig. 7 – A MRI scan volume of the lower limbs and a registered CT scan volume of left shank are visualised in the Global Slice View; in the lower View, the CT is shown at 0.5 transparency.

tional standard for medical imaging; input can also come from dynamic MRI, cardio CT, and other 4D imaging modalities. Polygonal surfaces can be imported from STL or VRML files. STL format, in particular, can be used to exchange geometry with CAD programs. Standard 8 or 24-bit 2D images can be imported from JPEG, TIFF, GIF and BMP files. 16-bit diagnostic images can be imported in DICOM format. Motion capture data can be imported from PGD files, from C3D files, or from raw ASCII files. C3D files are translated into a sub-tree list of moving landmarks, which can be later clustered into a segmental motion using a specific operation. PGD format, less popular but more sophisticated, already provides the segmental trajectory and is translated by the Data Manager into a moving cluster of rigidly linked landmarks for each segment in the file. Scalar signals such as ECG and EMG, as well as vector signals such as force platform signals can be imported from raw data ASCII files. Last, but not least, finite element meshes and results can be imported from Ansys neutral files.

Any other dataset, even if it is not supported by the Data Manager, can be imported—it is stored unchanged alongside the other data. These unknown data are usually processed with an operation called Open With External Program, where the correct application is launched on the basis of the file extension. Thus, we can add to our database an excel file containing all our calculations, which will be stored, along with the other data, in a single repository.

The effectiveness of creating databases of disparate data is dramatically extended by the ability of organising these VMEs into a hierarchical tree, the VME Tree. The pose matrix of each VME is interpreted as the pose of the VME with respect to the parent VME. In this way, it is possible to create very complex spatial chains of datasets, one referring to the other. With the notable exception of motion data, imported datasets do not contain information on the pose of the dataset with respect to a global reference system. Thus, the VME is created and attached to the root of the VME Tree, and an identity pose matrix is assigned to it. Afterward, the user can Cut&Paste or Reparent the VME under another VME. Cut&Paste leaves the pose matrix unchanged, while Reparent leaves the VME pose unchanged. In this way, we can relocate the VME in the VME Tree by preserving its local pose matrix, or its global pose matrix.

Specific operations can modify the pose of a VME by user-defined inputs or move it to a new pose in which the VME is aligned with another VME of the same type. Because of this separate pose matrix, each VME can be translated, rotated, and scaled independently from its type. We can apply spatial transformations to surface, volumes, and mesh grids in the same way. Those few Views whose performance would be drastically impacted by the fact that the dataset is arbitrarily oriented with respect to the View camera, are defined as Local, that is, they can render only one VME of that type at a time, and re-orient the camera to be always properly aligned with the VME reference system.

The entire VME Tree can be saved in the Multimod Storage Format (MSF). Currently, the directory structure of the MSF is based on the file system. Thus, saving a VME Tree creates a directory, not a file. Inside this directory, there is an XML file that defines the VME Tree structure and all metadata of the VMEs forming it, and various sub-directories containing

the raw data of each VME. By copying that directory and its contents to another location, we move the entire VME tree. We propose that zipped MSF files could become the de facto standard for exchanging biomedical data. The Data Manager has the advantage that it can not only open data contained in an MSF file and export it in different standard formats, as required, but that it can also rapidly visualize the data, for verification or other purpose, within the same environment.

Once disparate data are imported into the Data Manager, and organised into the VME Tree, they can be explored using the available Views. The basic idea is that each View provides a representation of different VME types that is internally coherent. So for example, in the Digitally Reconstructed Radiograph (DRR) View, any Volume VME is rendered as a radiograph taken from the current viewpoint, while a Surface VME is assumed to be a totally radio-opaque object, and is thus rendered with flat shading (Fig. 2). In the Arbitrary Slice View (Fig. 3), all displayed VMEs are sliced along a user-controlled plane; of course, each VME type requires a different slicing pipeline, but this is transparent to the final user.

We hope that this approach will overcome the complexity associated with choosing the right visualisation algorithm for the right data type—the View concept moves the visualisation to a higher level of abstraction, at which the user thinks in terms of the final representation independently of the type of data being processed. Each View also has its own interactor, which is coherent with the View type: Single-Pane Perspective Views allow pan, zoom and rotate using the mouse; multiplane multislice views, such as the OrthoSlice View using interactive gizmos to allow a finer control of the slices' position (see right bottom view of Fig. 5).

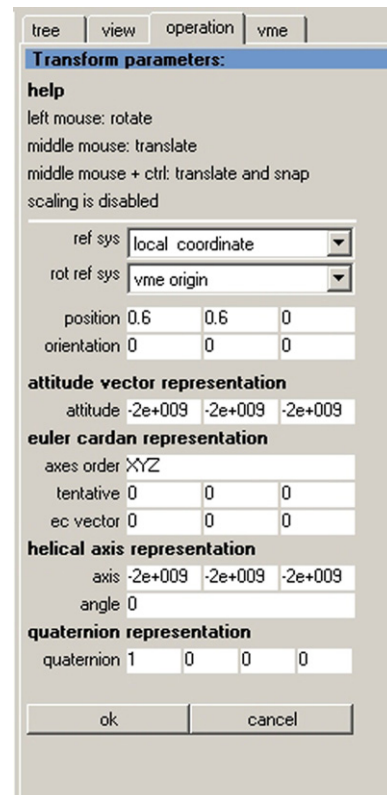


Fig. 8 – The Transform operation interface panel.

When we add a certain View type to the display, a check button appears on the VME Tree near each VME that can be displayed in that View. By checking and un-checking them, the user defines which VMEs should be visualised using that View. In this way, the user can add many instances of the same View type, each displaying a different set of VMEs (Fig. 4) or, alternatively, many different types of View, all displaying the same VME, but using different visual representations (Fig. 5). In practice, the user tends to create and destroy many views using various combinations of these strategies on a case-by-case basis.

Surfaces can be displayed using a Perspective Surface View, which supports multiple lights, “fly to” camera motion, and a multi-resolution scheme that ensures interactivity even with huge number of polygons.

Volumes can be visualised with various Volume Rendering Views. The Ray Casting View displays, in the same view, a CT or MRI volume, together with polygonal surfaces. The volume rendering transfer function is very sophisticated since it is multidimensional and is controlled by on-screen widgets that allow the characteristics of each individual tissue type to be adjusted interactively, with the effect being seen

instantaneously on the image. This allows the user to make visible even subtle differences, for example, to visualise muscles in a CT Dataset (Fig. 6). Another volume view is the Isosurface View, which allows the interactive extraction of an isodensity surface from a volume up to 512^3 voxels in size (see left bottom view of Fig. 5). The resulting isosurface can also be saved, thus providing also an initial segmentation tool.

The list is completed by simple Views to display 2D plots, 2D images, single volume slices, or complex composite views that combine various modalities into so-called Multimodal Display representations [13]. A special type of View is associated with data fusion; examples are the Global Slice View, which displays a single slice through multiple volumes, each in its own spatial position, and with its own colour clue (Fig. 7), or the RX-FEM View that merges CT data and finite element analysis results into a representation that provides an immediate visual understanding of the regions at higher risk of fracture (see Figs. 14 and 15).

Operations are commands that generate new VMEs or modify existing ones. Create operations create service entities such as VME groups, reference systems, or VME aliases, parametric

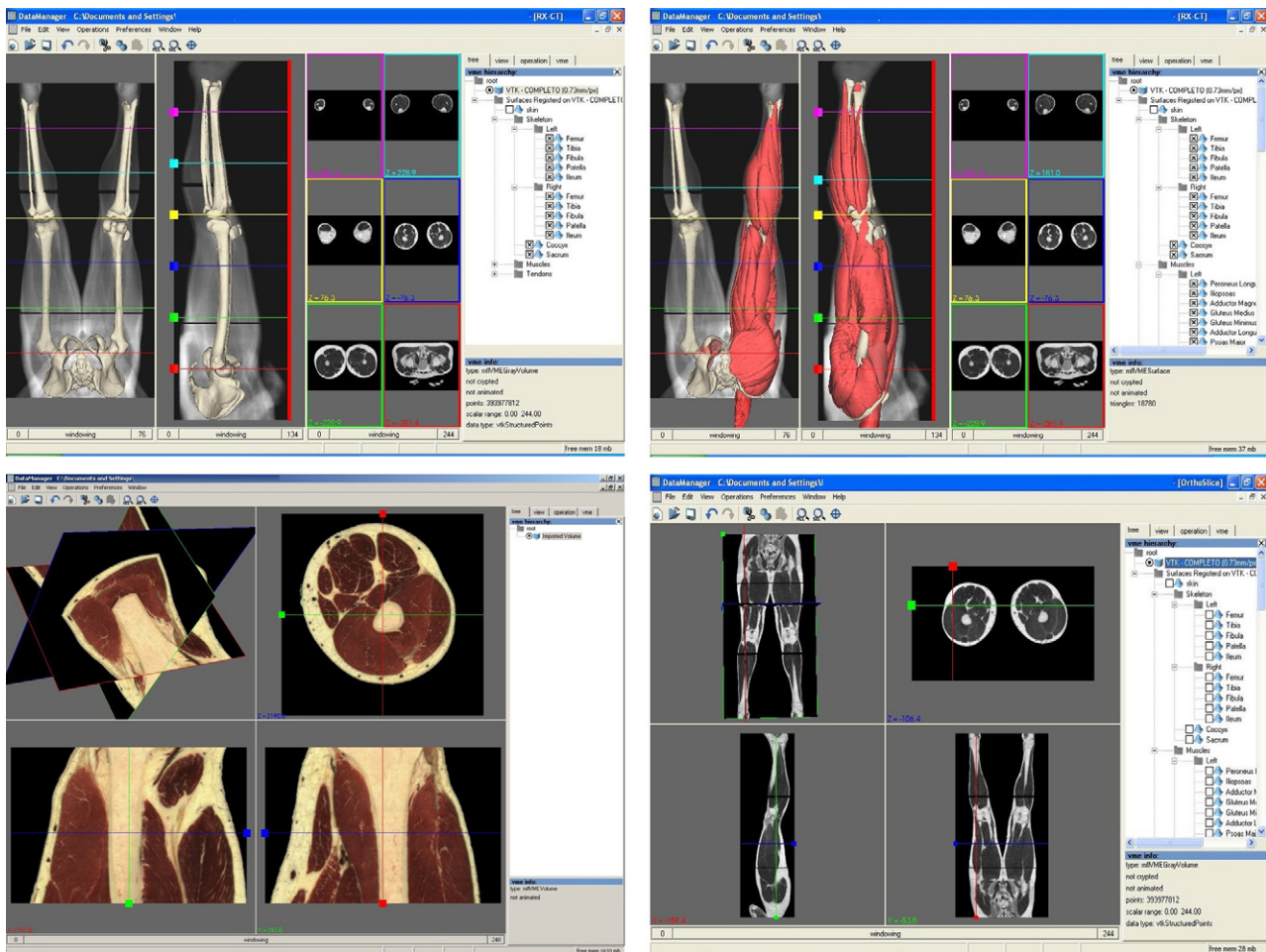


Fig. 9 – The CT dataset (top-left) is imported into Data Manager together with the bone surface: the labelled volume (top-right) obtained by segmenting the cryosection images is imported, registered to the CT dataset and then segmented to extract muscles and tendon surfaces; selected colour cryosections (bottom-left) and the MRI dataset (bottom-right) are also imported, registered and then used to control the accuracy of the segmented surface.

elementary solids (sphere, cube, cone, cylinder) or anatomical landmarks.

Derived operations add a so-called Derived VME to the VME Tree; these are special VMEs which result from the procedural manipulation of another VME. The procedural manipulation is called a data pipe. It can, for example, produce a 2D image posed in space by cutting a Volume VME with a Planar Surface VME; or create a clipped surface by clipping a Surface VME with a Parametric Solid VME; or dynamically regenerate the tubular surface that connects two landmarks, while they move in space relative to each other; or produce a coloured map representing the surface values on a FEM VME.

Modify operations can be used to add result sets to a Mesh VME, to add a note to any VME, to crop or resample a Volume VME, to group or ungroup landmark clouds, to filter polygonal surfaces, or to translate, rotate and scale any VME. Measure operations calculate the distance or angle between any 2D points picked on any View, or the average density with a volume of interest. Fuse operations register in space two landmark clouds, two polygonal surfaces, synchronise in time two time-varying VME, etc.

Some of these operations can be quite complex from an implementation point of view, while remaining relatively easy to use. The Transform operation applies a generic affine transformation to the selected VME, defined with

respect to any existing implicit or auxiliary reference system. After the transformation matrix is computed, a polar decomposition separates the roto-translation and the scaling components. The first pre-multiplies the VME pose matrix, defining the new pose; the second is applied directly to the VME data if scaling was intended or simply ignored if the transformation was intended to be rigid (in this case, the scaling is generated by round-off errors). The transformation can be defined not only in terms of matrix components, but also by expressing rotations in terms of cardanic angles, Euler angles, attitude vectors, quaternions, or helical axes. This produces a user interface for the Transform operation that is quite complex, but still manageable (Fig. 8).

Clever combinations of these operations can solve complex data fusion problems. By using the Virtual Palpation operation, which places new landmark objects on to a VME, it is possible to register an atlas of muscle insertions based on the Terry musculo-skeletal database [14] to a patient-specific skeletal anatomy. Motion data can be used to animate patient-specific bone models by registering the landmarks palpated in the motion-capture session with those virtually palpated on the bone models. Last, but not least, with virtual palpation, the user can define anatomy-based auxiliary reference systems, and use them to define spatial transformations.

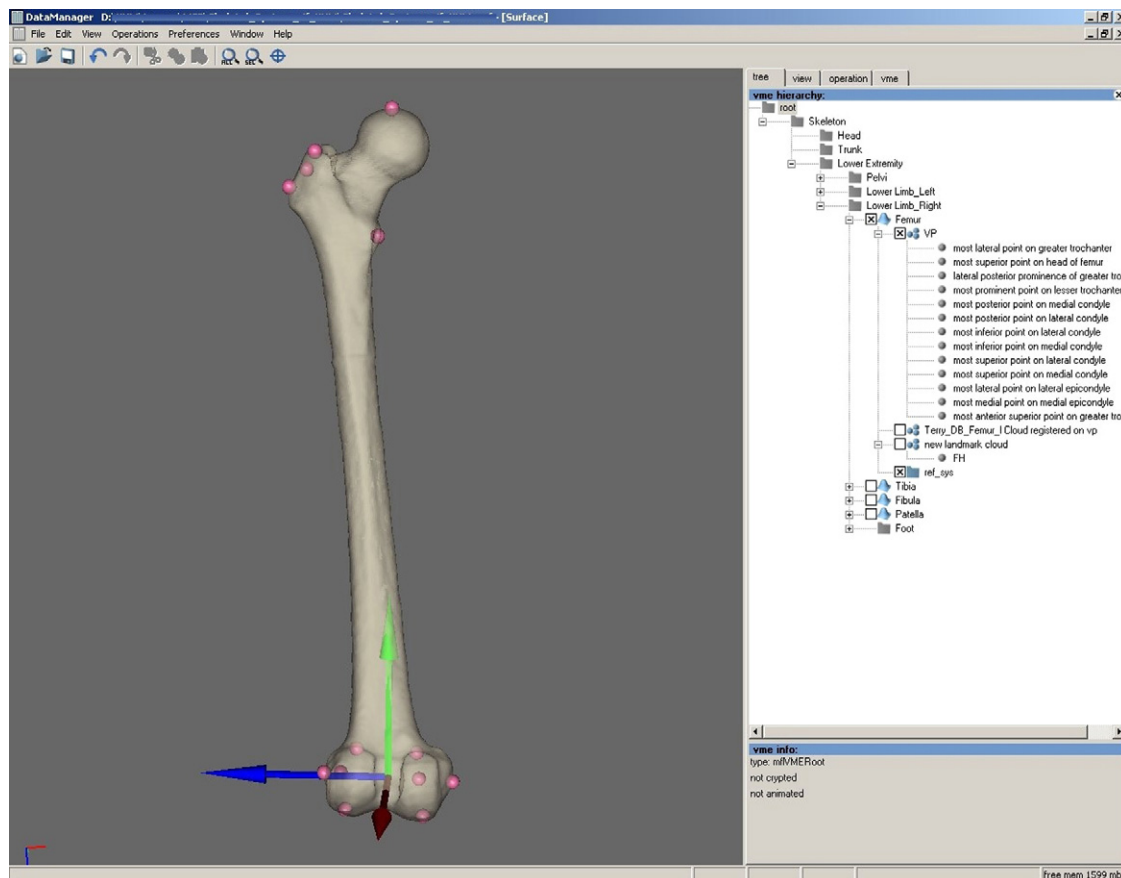


Fig. 10 – The left femur is virtually palpated to locate various skeletal landmarks, some of which are used to define an anatomical reference frame for the distal femur; knee movements are then described as the motion of this frame with respect to that of the proximal tibia.

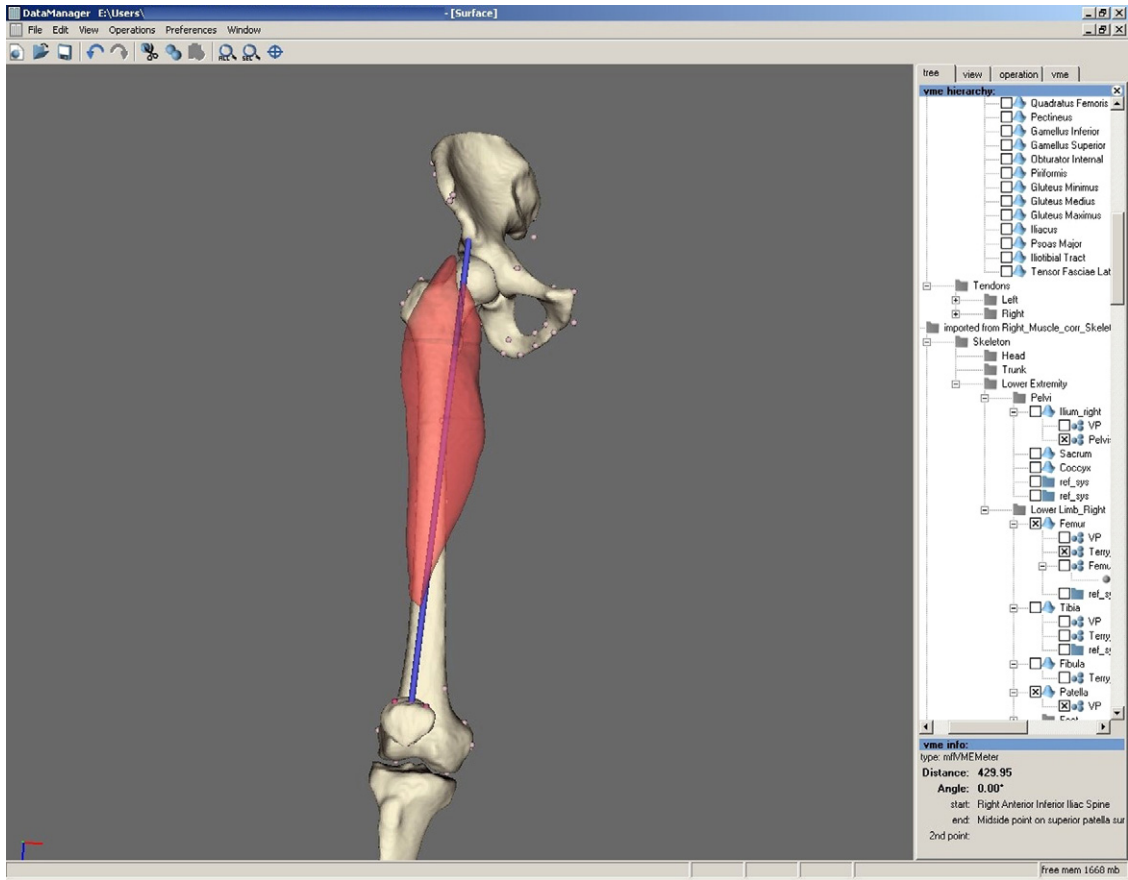


Fig. 11 – Determination of the biomechanical line of action for the Rectus femoris.

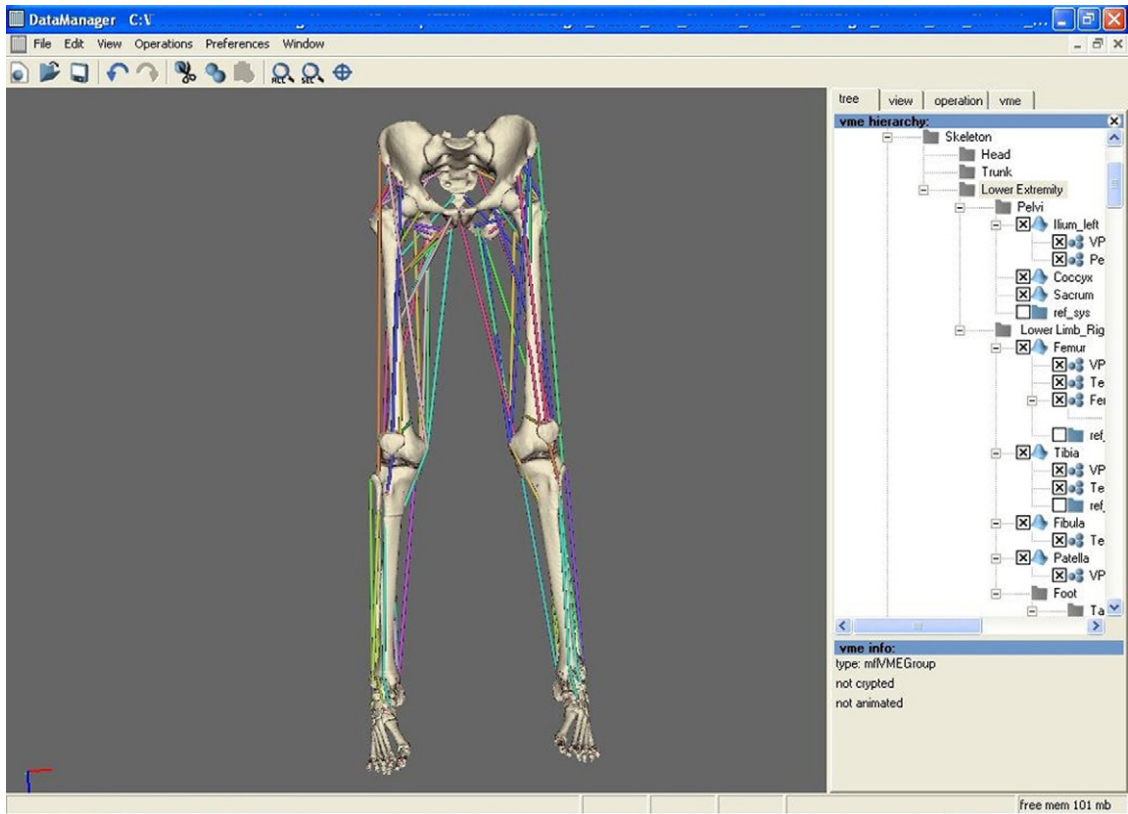


Fig. 12 – The complete biomechanical model of the Visible Human Male lower limbs.

3. Use case: biomechanical model of the Visible Human Male

The Visible Human Male is a collection of datasets generated by the National Library of Medicine as part of the Visible Human Project [15]. The corpse of an adult human male in previously good health was scanned for its entire length with MRI and CT, and then serial-sectioned using a cryogenic procedure. The body was sliced axially at millimetre intervals and a high-resolution digital image was taken for each slice. These datasets can be downloaded from the project web site.

All of the datasets relevant to the lower half of the body were loaded into Data Manager using the standard import capabilities—the DICOM importer was used for the CT and MRI data, while the cryosections were loaded as JPEG images.

Cryosection images were pre-processed with a segmentation application (AMIRA, Mercury Computer Systems, Berlin), and regions of muscle, tendon and bone were properly labelled. The labelled volume was then imported into Data Manager using the Raw Image Data importer. For each muscle and bone, surfaces were extracted from the labelled volumes using the Extract Labels operation; the surfaces obtained from this were then smoothed and decimated using the Filter Surface operation (Fig. 9).

All imported data were mutually registered in space using anatomical landmarks as reference points. Data Manager provides a powerful operation that allows the user to place landmarks in volumes and on surfaces. The same landmarks were located in each dataset, and the various landmark clouds were then registered with rigid transformations that were computed using the Register Landmarks implementing single-value decomposition. Alternatively, surfaces could have been registered using the Register Surface implementing the Interactive Closest Point algorithm. Direct volume-to-volume registration is currently being implemented in a parallel project [16].

Once all volumes and surfaces were aligned in space, the biomechanics modelling started. The bones forming the skeleton are represented in the CT, MRI and cryo dataset in the

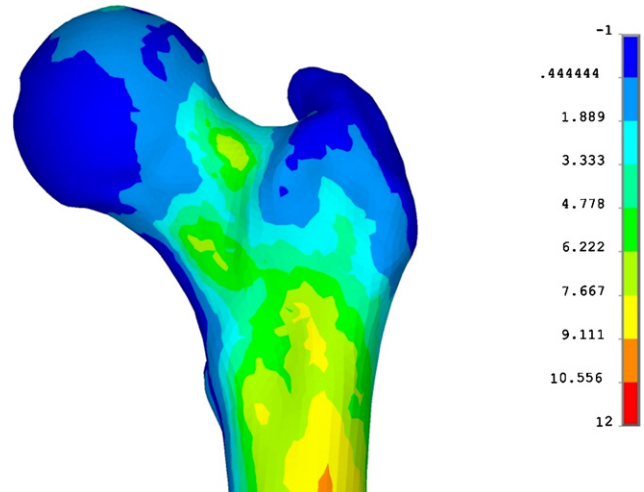


Fig. 14 – Principal tensile stresses in the proximal femur during level walking at normal speed.

position that is normal for a supine body (Fig. 9). Anatomical landmarks were located on each bone with the Add Landmarks operation, and anatomical reference systems were derived from them (Fig. 10) with the operation Create RefSys.

The lines of action of muscles were computed by fitting each 3D surface representing the muscle with one or more linear elements that were created with the operation Derive Distance Meter (Fig. 11). Landmarks were placed at the origins and insertions of these action lines. MRI images were used to define the separation between tendon and muscle, so as to compute the effective length of the muscle. Muscle volumes were calculated and these were used to compute the physiological cross-sectional area by evaluating the muscle volume divided by the effective length. A total of 86 muscle groups were considered in the biomechanical model (Fig. 12).

Spatial roto-translations were imposed with the operation Transform so as to align the reference systems of the bones with the configuration that is normally found in the standing posture (Fig. 13).

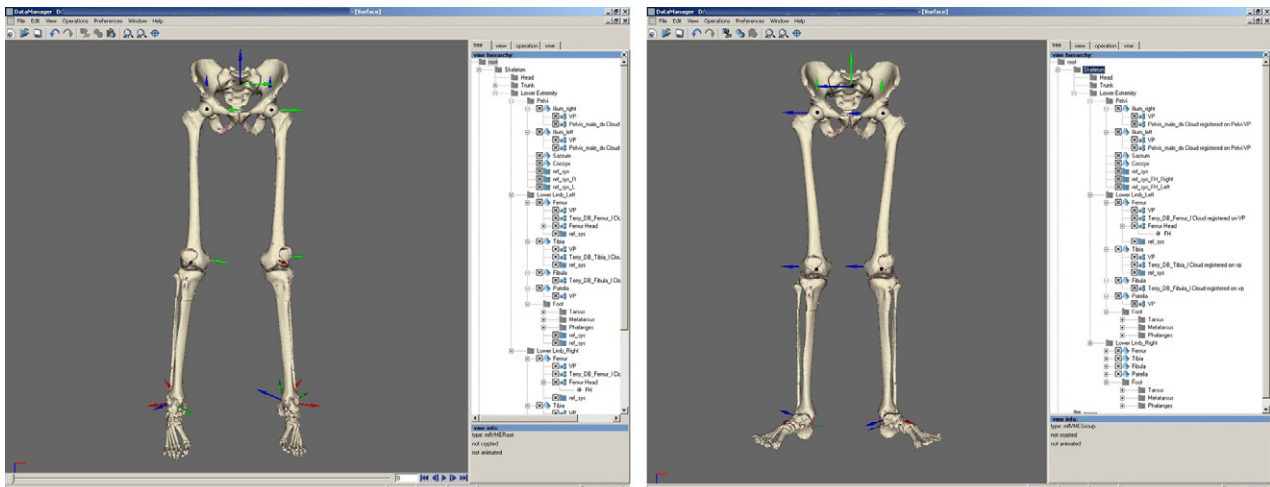


Fig. 13 – The lower body skeleton is supine in all VH datasets (left); joint rotations are imposed to align the bones with a conventional standing posture.

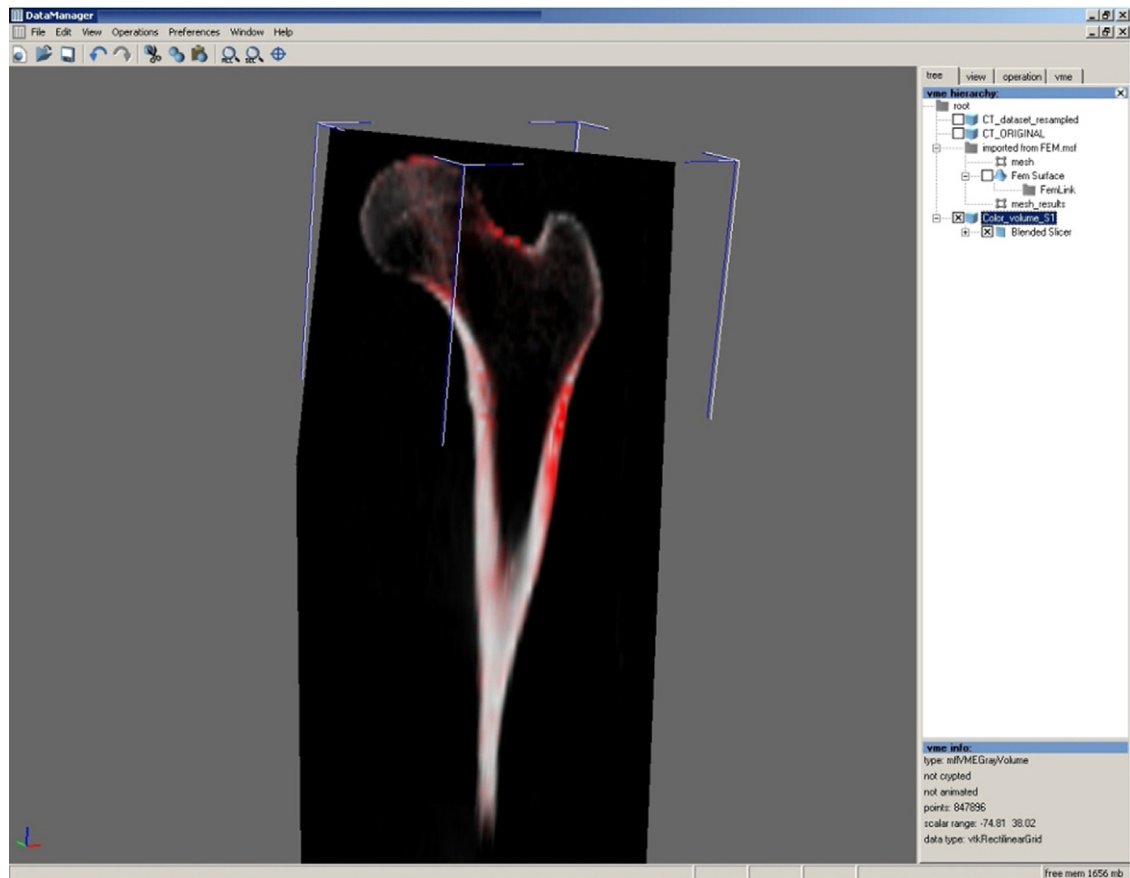


Fig. 15 – FE results are imported in the Data Manager, re-sampled on the CT volume grid, and visualised together with the CT images as a superimposed red clue that is more intense where the stresses are higher. Here, this multiscale volume is visualised using an Arbitrary Slice View. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of the article.)

Motion data relating to the body donor are not available. Thus, averaged data from a population of normal subjects, properly scaled for body size and weight, were used [17]. Level walking was represented as a temporal sequence of joint angles that were imported as raw motion data, and then associated with the skeletal model using the aforementioned landmarks/reference systems; this produced a walking skeleton. Since no ground reaction was available for the subject, the joint moments were derived from the literature as well [17].

All data relating to the instant known to produce the highest hip-joint reaction [18] were exported to a custom-made application that computes muscle and joint forces using the static optimisation procedure described in [19,20]. The femoral bone surface was exported to a pre-processing application (HyperWorks, Altair, USA) to create a finite element model. Muscle and joint forces were applied to the femur model, and the stresses and strains induced by these loading conditions were computed (Ansys 9.0, Ansys Inc., USA) (Fig. 14). The finite element mesh and the stress results were imported back into Data Manager and post-processed alongside the original medical images (Fig. 7). This was done by generating a synthetic volume colored with the scalars of the finite element results with the operation Color Volume, that was then visualized

superimposing it to the original CT dataset with the Blended Slicer operation (Fig. 15).

4. Conclusions

The present manuscript provides a general description of Data Manager, and of an illustrative use case. Other use cases are described in recent papers where Data Manager was used to solve disparate research problems [21–24].

From this brief example, it is possible to understand the potential of Data Manager. The current version of this application (version 1.2) can be downloaded from Biomed Town [25]. Documentation on the general concepts described here is now available [26], and support can be obtained from the newly created User Forum [27].

We believe the Data Manager can be used to solve a large number of problems commonly encountered in many biomedical engineering activities. Thus, we invite all the readers to freely download the software, to put it to use, and to write to the User Forum with all problems, needs and recommendations that may help to make Data Manager an even more useful tool in the future.

Acknowledgements

The authors would like to thank Luigi Lena for the illustrations and Mauro Ansaloni and Roberta Fognani for the support during the study. The authors would also like to thank all the people that contributed to the Multimod Project, including Alessandro Chiarini, Luca Astolfi, Stefano Perticoni, Silvano Imboden, Marco Petrone, Paolo Quadrani, Cinzia Zannoni, Alexander Savenko, Mel Krokos, Véronique Feipel.

REFERENCES

- [1] <http://www.avs.com/>.
- [2] <http://www.opendx.org/>.
- [3] <http://software.sci.utah.edu/scirun.html>.
- [4] <http://www.amiravis.com/>.
- [5] <http://www.ablesw.com/3d-doctor/index.html>.
- [6] <http://www.materialise.com/mimics/main.ENG.html>.
- [7] <http://www.slicer.org/>.
- [8] <http://homepage.mac.com/rossetantoine/osirix/Index2.html>.
- [9] <http://www.mayo.edu/bir/Software/Analyze/Analyze.html>.
- [10] <http://www.julius.caesar.de>.
- [11] M. Viceconti, C. Zannoni, D. Testi, M. Petrone, S. Perticoni, P. Quadrani, F. Taddei, S. Imboden, G. Clapworthy, The multimod application framework: a rapid application development tool for computer aided medicine, *Comput. Methods Programs Biomed.* 85 (2007) 138–151.
- [12] <http://www.openmaf.org>.
- [13] R. Lattanzi, M. Viceconti, C. Zannoni, P. Quadrani, A. Toni, Hip-Op: an innovative software to plan total hip replacement surgery, *Med. Inform. Internet Med.* 27 (2002) 71–83.
- [14] T.M. Kepple, H.J. Sommer III, K. Lohmann Siegel, S.J. Stanhope, A three-dimensional musculoskeletal database for the lower extremities, *J. Biomech.* 31 (1998) 77–80.
- [15] http://www.nlm.nih.gov/research/visible/visible_human.html.
- [16] Y. Yang, R. Mootanah, I. van der Linde, P. Ingle, P. Wou, M. Viceconti, The registration of CT-MRI images of biomechanical systems in the multimod application framework using mutual information and pixel intensity gradients, in: *Proceedings of the 2nd International Conference on Computational Bioengineering*, IST Press, 2005, pp. 1095–1103.
- [17] M.P. Kadaba, H.K. Ramakrishnan, M.E. Wootten, Measurement of lower extremity kinematics during level walking, *J. Orthop. Res.* 8 (1990) 383–392.
- [18] G. Bergmann, G. Deuretzbacher, M. Heller, F. Graichen, A. Rohlmann, J. Strauss, G.N. Duda, Hip contact forces and gait patterns from routine activities, *J. Biomech.* 34 (2001) 859–871.
- [19] J.H. Challis, Producing physiologically realistic individual muscle force estimations by imposing constraints when using optimization techniques, *Med. Eng. Phys.* 19 (1997) 253–261.
- [20] R.D. Crowninshield, R.A. Brand, A physiologically based criterion of muscle force prediction in locomotion, *J. Biomech.* 14 (1981) 793–801.
- [21] D.D. Anderson, J.K. Goldsworthy, K. Shivanna, N.M. Grosland, D.R. Pedersen, T.P. Thomas, Y. Tochigi, J.L. Marsh, T.D. Brown, Intra-articular contact stress distributions at the ankle throughout stance phase-patient-specific finite element analysis as a metric of degeneration propensity, *Biomech. Model. Mechanobiol.* 5 (2006) 82–89.
- [22] A. Leardini, L. Astolfi, S. Fantozzi, M. Viceconti, M.G. Benedetti, F. Catani, Advanced multimodal visualisation of clinical gait and fluoroscopy analyses in the assessment of total knee replacement, *Comput. Methods Programs Biomed.* 79 (2005) 227–240.
- [23] A. Leardini, C. Belvedere, L. Astolfi, S. Fantozzi, M. Viceconti, F. Taddei, A. Ensini, M.G. Benedetti, F. Catani, A new software tool for 3D motion analyses of the musculo-skeletal system, *Clin. Biomech. (Bristol, Avon)* 21 (2006) 870–879.
- [24] S. Van Sint Jan, X. Demondion, G. Clapworthy, S. Louryan, M. Rooze, A. Cotten, M. Viceconti, Multimodal visualization interface for data management, self-learning and data presentation, *Surg. Radiol. Anat.* 28 (2006) 518–524.
- [25] <http://www.biomedtown.org/biomed.town/B3C.Building/products/datamanager>.
- [26] <http://www.openmaf.org/maf/doc/maf-2-0-documentation/>.
- [27] <http://www.biomedtown.org/biomed.town/B3C.Building/products/datamanager/users.forum/>.