Network collaborative environment for human tissues 3D modeling

Přemysl Kršek, Michal Španěl, Miroslav Švub, Vít Štancl, Ondřej Šiler and Vítězslav Sára

***Abstract*— This paper deals with the new concept of network based virtual collaborative environment to support clinical applications of 3D models of human tissues, created from CT/MR data. It is a topic lying between 3D tissue modeling and PACS systems. Designed system allows clinical realizations of 3D applications as a service to clinical workplaces, provided by specialized 3D laboratory, even over great distances. Problem lies within the need of doing necessary consultations, corrections and verifications distantly. This is solved by our system in the form of virtual collaborative environment. This system is built upon three-layer client-server architecture. Our application is focused on 3D tissue modeling. Generally it can be used as a basis for other similar applications.**

I. INTRODUCTION

Recent trend in medicine has been individual approach to patient treatment. One of possible ways to achieve this, is 3D modeling of tissues for the needs of individual planing, simulations and navigation of performed surgeries.

Input information for 3D tissue models are image data obtained by computer tomography (CT) or magnetic resonance (MR). Today, CT/MR data are commonly available in standard digital format DICOM (Digital Imaging and Communications in Medicine). It has been more common to store obtained image data in PACS (Picture Archiving and Communication System) systems. These systems offer long-term archivation of stored data. When data are needed, these systems allow distant access to stored data and their transmission to another workplace via computer network.

Today, many various computer systems for 3D tissue modeling and support of clinical applications exist. All of them are running locally at one particular workplace. Process of preparing clinical applications with support of 3D tissue modeling consists of following steps :

- obtaining CT/MR data from PACS system
- tissue segmentation and 3D modeling
- preparation of clinical applications (planning, simulation, navigation)
- consultation, correction and verification of results

Everything needed for realization of particular clinical application is in one place: CT/MR data in PACS system, all concerned specialists (radiologist, surgeon, technician etc.), patient etc. Therefore, there are no technical difficulties in communication, data transmission etc.

However practically, all of conditions for realization of 3D applications are not to be found in one place. Only couple of big clinical workplaces can afford to operate their own workplace that is technically and personally capable of handling whole process of 3D application preparation. In rest of clinical workplaces, 3D applications are not used due to technical complexity, financial or personal difficulties.

One of possible solution to this problem is preparation of 3D applications by independent specialized workplace (3D lab). This lab would offer services to other clinical workplaces based on supplied CT/MR data. Problem arises when it is needed to consult, correct and verify prepared application. If 3D lab, clinical workplace and consultants are distant from each other, this concept of independent 3D lab is impractical.

This paper deals with problematics of virtual collaborative environment (VCE) suited for 3D modeling of human tissues based on CT/MR data. Basically it is an instrument that would allow consultations, corrections and verifications in virtual environment integrated by computer network. Participating specialists (radiologist, surgeon, technician etc.) would be able to cooperate even if in different buildings, cities or regions. This tool would make the concept of 3D lab realizing clinical application feasible. Current PACS systems do not offer this kind of services.

II. RELATED WORK

Recent five years we have been intensively participating on research and development in the field of 3D human tissue modeling and clinical applications. Recently we have been cooperating with several hospitals in Czech Republic (Saint Ann Faculty Hospital of Brno, Faculty hospital of Olomouc, Faculty hospital of Ostrava, Faculty hospital Motol Praha etc.) We have come across particular problems with distant communication whenever we need consultations and corrections of prepared 3D applications. When our cooperations were limited to local hospitals, it was no problem for us to gather at one computer desk. That is of course not possible for longer distance cooperation. Up to now, we have solved this communication problem off-line. Data were passed through network via SFTP servers or PACS systems. Resulting iterative process of consultations, corrections and verifications was lengthy and not intuitive.

For 3D modeling of tissues based on CT/MR data we have developed our own graphical user environment MediTools (Fig. 1). This system works locally and provides full 3D visualization of CT/MR data. (volume rendering, multiplanar cross sections, etc.) and user support of tissue 3D modeling. 3D visualization is based on C++ library *OpenInventor*. Data core and algorithms for volume data processing is provide by MDSTk library [2]. 3D tissue modelling process consists of following steps:

Autors are with Faculty of Information Technoly, Brno University of Technology, Božetěchova 2, 612 66, Brno, Czech republic, krsek/spanel/svub/stancl/siler@fit.vutbr.cz, xsarav00@stud.fit.vutbr.cz

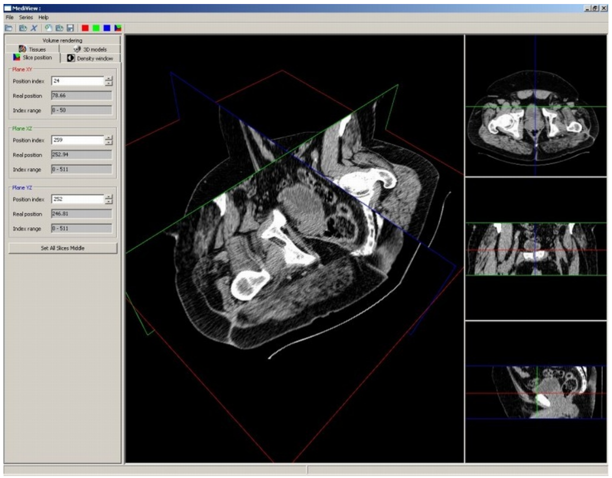

Fig. 1. System MediTools graphical user interface.

- Import of CT/MR data in DICOM format
- Automatical tissue segmentation with manual correction of results [1] [2]
- Automatical creation of 3D models of selected segmented tissues (Flow Reduction Marching Cubes) [3] [4]
- Smoothing of generated 3D tissue models (Geometric signal processing on polygonal meshes) [5] [6]
- Decimation and smoothing of models (Surface simplification using quadric error metrics) [7] [8]
- Preparation of individual clinical applications based on created 3D tissue models [9] [10]
- Consultation, correction and verification of results

Our clinical applications mainly involve these fields: stomatology, plastic surgery, traumatology, orthopaedy and neurosurgery. We have recently been cooperating on 40 realized cases : 14 cases of tooth autotransplantation, 21 cases in orthodontia, 11 cases in traumatology/orthopedy, 3 cases in plastic surgery and 1 case in neurosurgery (Fig. 2).

III. NETWORK COLLABORATIVE ENVIRONMENT

For solving communication problems with consultations and corrections of prepared 3D applications between distant workplaces we have been trying to add possibility to work in virtual collaborative environment (VCE) to our current MediTools system. For this case, we have two possible system architectures, Peer-to-Peer and Client-Server.

A. Peer-to-Peer architecture

3D visualization in MediTools system is based on C++ library *OpenInventor*. This we have used together with *OpenVCE* library [11] and it allows us to automatically synchronize state of whole *OpenInventor* based 3D scene. OpenVCE library implements network Peer-to-Peer communication based on TCP/IP via SSL sockets. Extending our current MediTools system with communications in VCE framework is relatively simple. It comprises linking *OpenVCE* library and ensuring synchronization of additional data, which are not part of 3D *OpenInventor* scene.

VCE realized this way on local network was running with no serious problems up to five concurrently connected users. When more users were connected or communication was realized outside local network, network delays were noticeable.

Due to character of communication, initial state synchronization of VCE was problematical. This was because the system does not comprise central data storage and providing access to data to everybody. Individual changes (off-line) of users were local and had to be initially distributed to all participants. This scheme was vulnerable to conflicts in different versions of data being processed.

The most serious problem was however communication between 3D lab and clinical workplaces, usually part of hospitals. IT security policy of these hospitals is generally ”paranoid” so it maximally tries to limit whatever non-standard communication outside these. Our effort to run Peer-to-Peer communication outside hospitals over SSL sockets was strictly rejected. Only acceptable way was standard HTTP/HTTPS communication with secure and certified server outside hospital.

B. Client-Server architecture

It is for these reasons we switched to three-layer architecture Client-Server for further VCE design and development. (Fig. 3). VCE clients will communicate through central server using HTTPS protocol running on default port 80. This will take out the main problem of IT security policy of hospitals. It is also not a problem to detach VCE communication by redirecting the connection to private reserved port, via reserved line or VPN connection, etc.

1) Central server concept: Central server is built on three-layer architecture. HTTPS communication is provided by standard WWW server. Application layer consists of group of small specialized modules, implemented according to the needs of performance in PHP (database, storage) or C++/FastCGI (VCE). Development of whole system can be therefore focused on VCE solution itself. For remaining parts of server solution (database, storage, network communication, HTTPS protocol etc.), commonly used, debugged and secure products (Apache, MySQL, PHP, Linux, HTTPS encrypting etc.) are going to be used. In case where there is a demand to deploy the whole system in security and reliability critical conditions, it is no problem to switch to certified solutions of reputable firms (Sun, IBM, Oracle etc.).

Thanks to the integration of patient database and CT/MR data repository on central server, the problem of synchronization, archivation and version administration is automatically solved. With central server, it is possible to work locally. Client will be able to download current data of selected patient, do the whole process of 3D application preparation off-line and upload results afterwards.

2) VCE server: All of the source CT/MR data and prepared 3D applications are stored on server. At the moment when it is needed to consult or verify them, it is enough to create a VCE instance on server and upload the data of selected patient. Involved clients can then access this running

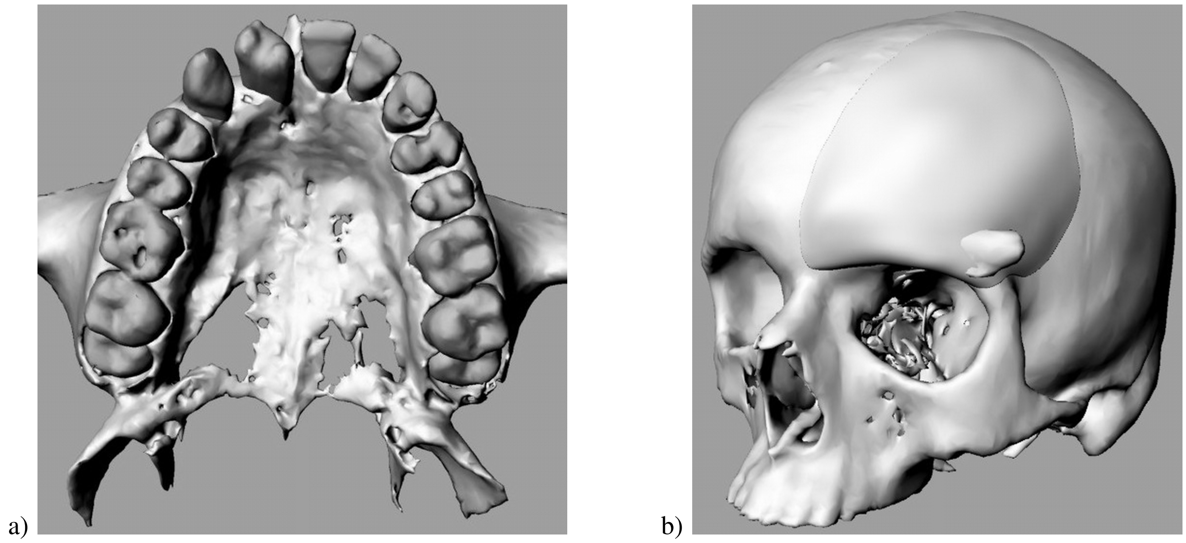

Fig. 2. 3D applications example: a) Maxilla and teeth 3D model for tooth autotransplantation simulation, b) Skull 3D model for plastic surgery face correction planning.

VCE instance and work with the data together: viewing and modifying, showing areas of interest to each other, etc.

VCE on central server is represented by several running processes, which administrate their own copy of data currently being processed and queues of incoming and outcoming requests. All valid incoming requests for change in current VCE data are immediately replicated into working copy on server, which represent the current state of VCE. All connected clients are synchronized according to this VCE state. Normally, clients themselves synchronize the state of their local data copies according to requests of changes and ordered according to how they are stored in queues on the server. If for any reason a delay of one client occur, for example server timeout, this client downloads current state of data from VCE working copy on the server.

Thanks to multitasking character of working with WWW server, VCE itself multitasks. With one VCE, more clients can safely communicate with. There can be also several VCEs running concurrently. As requests number and number of connected clients rise, or number of concurrently running VCEs, it is not a problem to scale-up the capacity or performance, thanks to its modular architecture.

Larger volumes of transmitted data occur only at the beginning, when clients are logging to VCE and synchronizing current data states. Largest dataflow is due to CT/MR data transmission, which are for our applications supposed to be read-only. Data peak involved in working with VCE can be effectively lowered by caching data already read to a local hard-drive with identification with their hash code. When working with VCE itself, amount of transmitted data is relatively small in order of 100 B $\sim$1 kB/s.

The biggest general problem of collaborative systems is delay of computer network. Due to the fact, that we need to communicate over generic internet network, no connection parameters are guaranteed, which could affect the fluency of work with the system. This problem we are going to solve by weakening the requirements on VCE. Our application are specific. We do not need to explore virtual world that is shared by the clients. We therefore do not need to solve conflicts of time continuity of generated changes, so that clients could "argue" over a particular object in 3D scene (changing of view, changing the position of objects, changing the data, etc.). We have therefore introduced a rule, which would ensure, that only one client at time (master) can generate changes to VCE. Other clients (slaves) are being notified after last synchronization. Master role is to be passed by token request system.

3) VCE client: Our MediTools system is to be used as graphic client. It is basically a "fat" client, which is in possession of its copy of processed data. Clients themselves ensure visualization, interaction, changes in data etc. The only thing server ensures is synchronization of data and parameters of their visualization. Connection to server is invoked by client, which passes its requests and awaits answer. Clients periodically post requests on whether there are new changes in data state. If so, they perform synchronization and data replication into their own copy. According to requests, multiple versions of VCE clients can exist. Simpler versions will go as far as visualizing current data state, higher versions will allow generating of changes in data.

Concurrently to working with VCE, it is useful to utilize other means of communication as Skype, NetMeeting, VOIP etc. Rights to access the system are administered by WWW server. Access to stored patient data and their CT/MR data is administered on SQL database level.

4) Implementation: Current version of graphical clients of MediTools system has been developed in C++ and for visualization, it utilizes graphical library OpenSceneGraph. Communication server is based on Apache WWW server, client database is MySQL. Application server is assembled according to performance requirements from PHP or C++/FastCGI scripts.

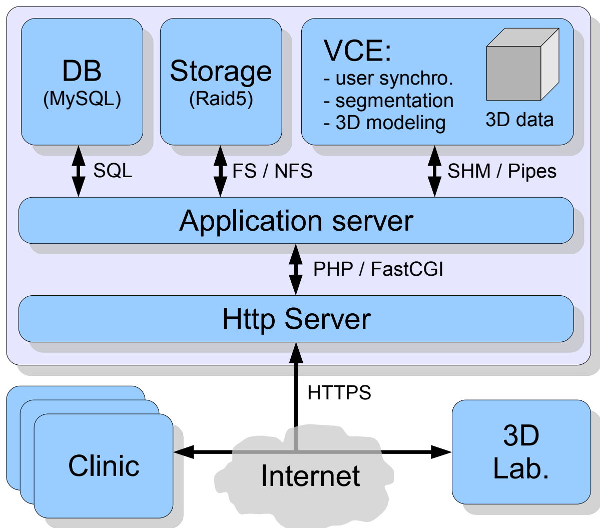

Fig. 3. Our network virtual collaborative environment system scheme.

Currently we have the main infrastructure of whole system implemented: communication HTTP/HTTPS library with C++ interface for clients, VCE application modules icluding unified binary and text data coding; PHP application modules for communication with client database and their CT/MR data; basic VCE functionality involving parameter of 3D scene visualization synchronization (camera movement, multiplanar slices movement, density parameters of visualization of CT/MR data etc.)

5) Testing: We tested our current implementation of our VCE system in simulated conditions. 4 to 6 clients were concurrently connected with only one active VCE instance. Clients were to be found in different cities of Czech Republic, Brno, Ostrava, Olomouc and Prague. All of test clients were connected through generic internet network on premises of a university, which mean connection of rather high quality.

Server was running on following configuration: 2xIntel Xeon 2,1 (dual core), altogether with 4 GB RAM, Raid 5 disc array 4,3 TB. Server was connected to a Gigabit backbone. Server is located at FIT BUT.

No significant network delay occurred in specified conditions, and volume of transmitted data was also not a problem. We have to keep in mind however, that it is only partial implementation, which performs data synchronization of 3D visualized data only. Detailed network load evaluation and delay of responses measuring we are not planning to do, until we have the whole VCE implementation ready, including tools for data editation.

IV. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

Based on wide practical experiences with clinical applications of 3D modeling in medicine, we have designed computer system for Virtual collaborative environment. This system allows to do consultations, corrections and verifications of prepared 3D clinical applications through generic computer network. Its main concept is maximum usage of current technologies (HTTPS, PHP, FastCGI, MySQL, etc.) in the form of three-layer application.

Designed system is generic enough to be used in many other fields, where it is necessary to work within VCE. Up to today, we have done the implementation of basic functionality of designed system (data archiving, visualization synchronization).

To get rid of the problem of network delay, we have limited our VCE system to allow data change from only one client at time, which is suitable for this type of cooperation. Even when communicating with couple of clients across whole Czech Republic, the delay has not had significant affect on the fluency of work with VCE. Data traffic is also at very low rate, with one exception: the moment when initial data synchronization is being done.

B. Future Works

- Implementation of additional VCE functionality
- Digital signature on imported CT/MR data
- Digital signature on changes
- Implementation of change workflow
- Archivation of changes

V. ACKNOWLEDGMENTS

Supported by grand project No. MSM6383917201 CZ and MSM0021630528 CZ

REFERENCES

[1] Bankman I. N.: Handbook of medical imaging: Processing and Analysis, Academic Press. Orlando, FL, USA Inc., 2000, ISBN 0-12-077790-8
[2] Španěl M., Kršek P.: Vector-based Medical Image Segmentation using Adaptive Delaunay Triangulation, In: Proceedings of the Sixth IASTED International Conference on Visualization, Imaging, and Image Procesing, Palma de Mallorca, ES, ACTA Press, 2006, p. 6, ISBN 0-88986-600-7
[3] Lorensen W., Cline H. Marching cubes, A high resolution 3D surface construction algorithm. In Siggraph'87 Conference Proceedings, pages 163-169, 1987
[4] Kršek P.: Flow Reduction Marching Cubes Algorithm, In: Proceedings of ICCVG 2004, Warsaw, Springer, 2005, p. 100-106, ISBN 1-4020-4178-0
[5] Balendran B., A Direct Smoothing Method For Surface Meshes: Proceedings, 8th International Meshing Roundtable, South Lake Tahoe, USA, 1999, 189-193
[6] Taubin G., Geometric signal processing on polygonal meshes: Eurographics 2000 State of The Art Report(STAR), September 2000.
[7] Schroeder W. J., Zarge J. A., Lorensen W. E. Decimation of triangle meshes. In Siggraph'92 Conference Proceedings, pages 65-70, July 1992.
[8] Garland M., Heckbert, P. Surface simplification using quadric error metrics. In Siggraph'97 Conference Proceedings, pages 209-216, 1997.
[9] Černochová P., Kaňovská K., Kršek P., Krupa P.: Application of geometric biomodels for autotransplantation of impacted canines, In: World Journal of Orthodontics 2005, Paris, FR, 2005, p. 1, ISBN 1530-5678
[10] Krupa P., Kršek P., Černochov P., Molitor M.: 3-D real modelling and CT biomodels pplication in facial surgery, In: Neuroradiology, Berlin, DE, Springer, 2004, p. 1, ISBN 0028-3940
[11] Peiva Jan: Active Transaction Approach for Collaborative Virtual Environments, In: ACM International Conference on Virtual Reality Continuum and its Applications (VRCIA), Chinese University of Hong Kong, HK, ACM, 2006, s. 171-178, ISBN 1-59593-324-7