Automatic Creation of 3D Environments from a Single Sketch Using Content-Centric Networks

Theodoros Semertzidis, Informatics and Telematics Institute and Aristotle University of Thessaloniki
Petros Daras, Informatics and Telematics Institute
Paul Moore, Atos Research & Innovation
Lambros Makris, Informatics and Telematics Institute
Michael G. Strintzis, Informatics and Telematics Institute and Aristotle University of Thessaloniki

ABSTRACT

In this article a complete and innovative system for automatic creation of 3D environments from multimedia content available in the network is presented. The core application provides an interface where the user sketches in 2D the scene that s/he aims to build. Moreover, the GUI application exploits the similarity search and retrieval capabilities of search-enabled content-centric networks to fetch 3D models that are similar to the drawn 2D objects. The retrieved 3D models act as the building components for the automatically constructed 3D scene. Two CCN-based applications are also described, which perform the query routing and similarity search on each node of the CCN network.

INTRODUCTION

In the new era of the *media Internet* where multimedia content is overflowing the networks and users are usually near a networked device, the available forms of communication seem inadequate. IP has changed the way we communicate and entertain ourselves, using voice over IP (VoIP), videoconferencing, IPTV, email, instant messaging, social networks, and so on. However, now is the time users really need a completely new user experience that transcends these communication forms.

In this new form of communication, 3D virtual environments will be the places for meeting, conversation, and entertainment among friends and colleagues. Moreover, these places will be constructed dynamically from each user based on the exact needs of the communication session. In order to do so in an easy and intuitive way, users should have the ability to search, retrieve, and use the deluge of multimedia content available in the networks as building components of a new 3D environment. However, in the current Internet architecture search and retrieval of multimedia content is not addressed sufficiently since the need for such services appeared rather late in the design process. On the other hand, the new trend of content-aware networks aims to solve exactly theses issues by focusing on the content, not on the host computers and their network addresses. This approach allows the user to retrieve content by its name without knowing where this content resides.

The Internet has evolved from a network of computers to a limitless resource of multimedia content with critical societal and commercial impacts. This fact led the academic community to study different new architectures that put the content in focus. In the 4WARD project a system for flexible, modular network components is examined [1], and an information-centric paradigm focuses on information objects rather than end-to-end connections. In the ANA framework generic abstractions of networking protocols are presented in order to support network heterogeneity and coexistence of clean slate and legacy Internet technology and protocols [2]. Koponen *et al*. propose a replacement of DNS with flat, self-certifying files in the DONA architecture [3]. Van Jacobson *et al*. [4] proposed the content-centric networking (CCN) approach, where the content is routed based on hierarchically named components. The CCN protocol is based on two packet types, Interest and Data. The consumer transmits an Interest packet, which is propagated in the network only to the nodes with available content. As a result, Data packets return to the consumer from the path the Interest packet passed. Based on the CCN architecture, Daras *et al*. [5] proposed an extension that introduces similarity multimedia content search in such networks. In search-enabled CCN, the user is able not only to retrieve content objects by name but also query the network for similar multimedia content.

At the same time, there are a plethora of academic as well as commercial attempts to build a workflow that target on virtual 3D environments. Many of them aim at corporate communications,

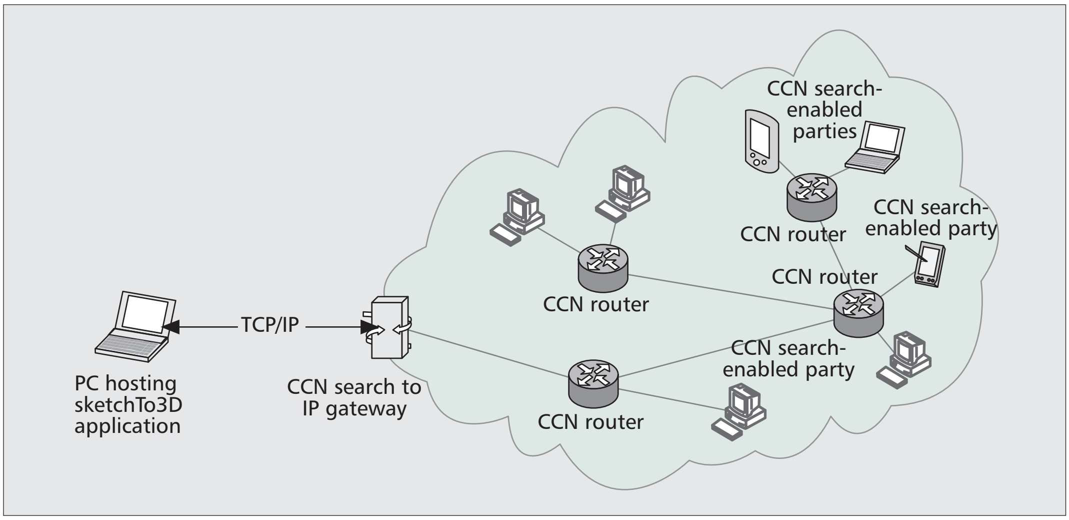

Figure 1. *The system's architecture.*

Querying by example (QBE) is a very common technique for querying multimedia databases. However, a prerequisite for the user is to have an object that is similar to what s/he is looking for. From this point of view, the QBE approach may not be that practical for a single-modality search.

events, and meetings, while others aim at e-learning applications or entertainment.

Products such as Assemb'Live, 3Dxplorer, web.alive, and Second Life are some of the commercial software available as services or appliances. All of them have their advantages and disadvantages based on the targeted audience due to the different background of each firm. Unfortunately, none of them provide a fast and easy-to-use interface for constructing on the fly a 3D environment from the media content that exists on the network nearby.

Sketch as a user interface has been studied in various works in recent years, since drawing is one of the primary forms of communicating ideas. Humans tend to draw, with simple feature lines, the objects or places they want to describe. Chen *et al*. use sketch input along with some word tags to retrieve relevant images from a database and montage a novel image [6]. However, the search queries are in fact word tags, while the sketch helps only with the positioning of the objects and refinement of the search results. The PhotoSketch system presents another approach for achieving sketch to image retrieval, based on the direction of the gradients of the sketch images [7]. Pu *et al*. [8] present a sketch-based user interface for retrieving 3D computer-aided design (CAD) models. Various papers propose different descriptors and methods for extracting features from sketch images for 2D image or 3D model retrieval. However, to the best of our knowledge, none of them is targeting the creation of 3D worlds using such asymmetric retrieval of 3D models from 2D sketches.

In this work we present a complete working environment for automatic construction of such 3D virtual places by exploiting the search-enabled CCN architecture and applying multimedia retrieval principles for the retrieval of 3D models from 2D sketches. The user sketches a 2D drawing of the scene s/he wants to construct. The objects, drawn in the 2D sketch, form the queries dispatched in a CCN to find similar 3D models. The similar media content are retrieved from the CCN through a gateway, and, finally, a 3D environment is automatically constructed.

The rest of the article is organized as follows. The next section presents the overall system architecture and the subsystems of the proposed framework. We then give a detailed description of the proposed sketchTo3D application. Next, we describe the search components of the CCN network and present the experimental setup. The final section draws conclusions on the current work and gives insights for the future.

System Architecture

The main components of the proposed system are an application named *sketchTo3D*, which provides the sketch user interface and displays the 3D virtual environment; the searchGateway application, which acts as a gateway between TCP/IP and the CCN network; and finally, the searchProxy application that runs on every search-enabled CCN party in order to perform a similarity search in their local repository.

Figure 1 presents the overall architecture with the search-enabled CCN, the CCN search gateway, and the end user's PC that hosts the sketchTo3D application. All these components are explained in detail in the following sections.

The sketchTo3D Application

Querying by example (QBE) is a very common technique for querying multimedia databases. However, a prerequisite for the user is to have an object that is similar to what s/he is looking for. From this point of view, the QBE approach may not be that practical for a single-modality search. On the other hand, in a multimodal setup where objects from one modality are always available to the user (e.g., sketch) for the formation of queries, the searching in such a system grants a greatly enhanced user experience.

The sketchTo3D application aims to provide the user with an easy-to-use interface for searching 3D models and building a 3D virtual world.

The application is coded in C++ using the QT 4.6 library for the graphical user interface (GUI), multithreading, and networking components. For the rendering of the 3D models, the openGL library is used.

For the insertion of a 3D model into the 3D virtual world, three basic parameters should be known: the positioning of the object in the 3D space, the scale of the object with respect to the other objects of the scene, and the orientation of the 3D model in order to view it from the right angle.

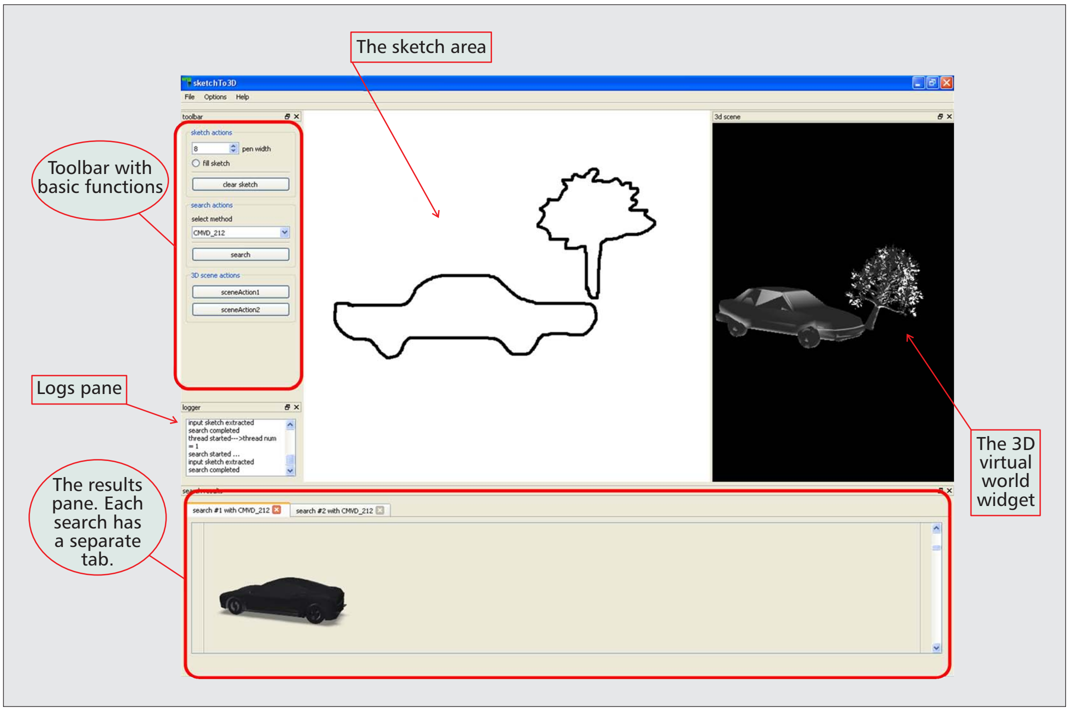

Figure 2. *The GUI of the sketchTo3D application.*

The sketch area is locked at the center of the GUI, and all other widgets are flexible to be docked at the left, right, top, or bottom of the application allowing the user to build a working environment that fulfills his/her needs (Fig. 2). At the left pane of the GUI (as depicted in Fig. 2) there is the basic toolbar with actions such as brush size or clearing the sketch. The submit button is pressed every time the user finishes drawing an object in order to start the search for similar 3D objects. The search process runs on a separate thread; thus, the user can continue drawing the next object in his/her scene.

To the right side of the GUI the 3D virtual world is depicted as it is seen from a single camera pointing at (0,0,0) in Cartesian coordinates. The user may translate, rotate, and zoom the virtual camera by applying left, middle, or right click-and-drag mouse gestures, respectively, on the widget.

Finally, at the bottom of the GUI the tabulated widget for the presentation of the search results is locked. Each search session is separated in a different tab in which the results are presented in ranked lists from the most to least similar objects.

By double clicking on an object in the results pane, the 3D model is placed in the scene at the correct position. When all the results have been retrieved and the user has selected the appropriate 3D models from the ranked lists, the 3D environment is completed. By applying the same mouse actions described above, the user may navigate inside the 3D world and explore the object set.

Submitting a Search Query

After the user clicks on the search button of the toolbar, a new search session is initiated. Since the search has to be conducted on each object of the scene alone, a segmentation step has to take place first. Our segmentation technique makes the assumption that the user does not draw overlapping objects. Taking this into account the application keeps a history of the sketched images between each search. For isolating the new object from the scene, a simple subtraction of the current sketch image from the previous one is automatically made. Then a bounding square of the object is found. If the bounding square is larger than 100 × 100 pixels, a scaling step takes place in order to have a 100 × 100 pixels image of the object. The procedure is presented in Fig. 3. The scaling of the query image helps the descriptors extraction algorithm run faster, and thus be efficient for real-time usage, without crucially affecting the retrieved results.

After the isolation of the sketched object from the scene, a new thread is initiated that handles all the procedures concerning this search without blocking the whole application from working properly. The query image extracted from the segmentation process is fed in the CMVD descriptor extractor [9], and a descriptor vector of low-level features of size 212 is computed. This descriptor vector is the actual query for each search. The next step is to make a request for connection to searchGateway using a TCP socket. When the connection is established and after a handshake process, the descriptor vector is sent to searchGateway for searching the CCN network. Finally, the thread enters in an idle state waiting for the gateway to reply with results.

From Sketch to 3D

For the insertion of a 3D model into the 3D virtual world, three basic parameters should be known: the positioning of the object in the 3D space, the scale of the object with respect to the other objects of the scene, and the orientation of the 3D model in order to view it from the right angle. The following subsections describe our approach to these issues.

3D Model Positioning — The position of the 3D object in the 3D environment has to be inferred from the position of the object in the 2D sketch image as well as with respect to other objects in the sketch.

In our approach we assume that all the objects of the scene are attached to the ground, so the Y coordinate of the 3D world is always Y = 0 (we have no elevation). Thus, by using as a reference point the lower-most point of the sketched object, we map the X-Y coordinates of the 2D to the X-Z coordinates of the 3D world, as seen in Fig. 4.

A heuristic stretch factor needs to be taken into account in order to map the height of the sketch image to the required depth in the 3D world. The user interface provides the user the ability to change the stretch factor and experience a deeper or narrower 3D scene, after the insertion of the objects in the scene (Fig. 2).

Scale — Taking into account the fact that all the 3D models are normalized at the unite sphere as in [9], the ratio of the 2D sketches may be used to apply the scale in the 3D environment. Indeed, the sketchTo3D application keeps an array of all objects that are drawn in the 2D sketch and their relevant sizes based on the size of the first object drawn. Moreover, in order to keep the ratio intact, we calculate the relevant sizes with a bounding square of every object at its centroid. Finally, based on this information a scaling factor is applied to every 3D model that is entered in the 3D virtual environment.

Orientation — The orientation of 3D models is still an open issue, and much research is conducted toward an efficient solution. However, for this application we use a simple solution that works efficiently.

We make the assumption that if we have a large unbiased database of 3D objects and their 2D view descriptors, and query this database with a sketch of a 2D view, the most similar results would be not only the most similar objects, but also the exact views of these similar objects.

Moreover, by using the rotation matrix we used for the extraction of the 2D views of each model, we may orient the 3D models in the 3D world with respect to the virtual camera coordinates. Since the descriptor extraction technique we use extracts descriptors for 18 views of each 3D model, the application is able to orient the 3D model in question in 18 different angles in the scene [9]. For finer tuning of the orientation of the 3D model, the user interface of the SketchTo3D application allows for manual in-place rotation of the model.

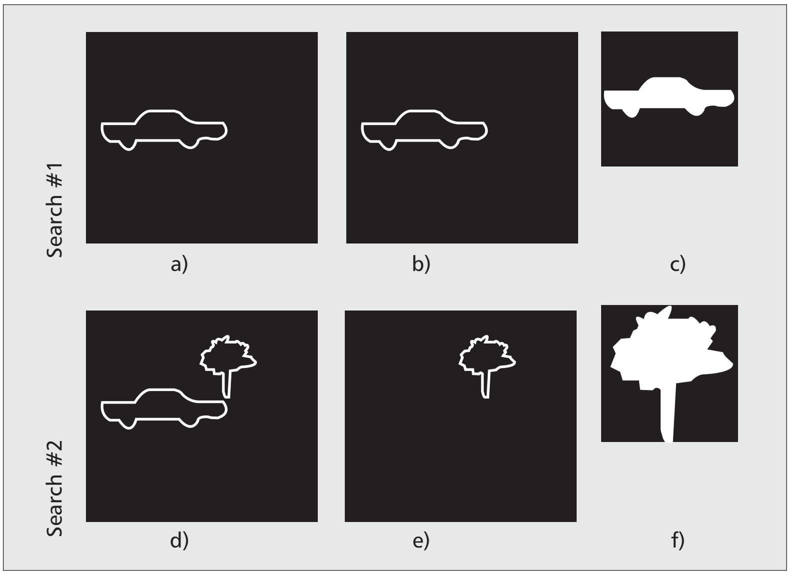

Figure 3. *The steps for the extraction of the query image from the sketch: a) 800 × 745 pixels. The current sketch and the image to subtract for the next sketch; b) 800 × 745 pixels. The Isolated object; c) 100 × 100 pixels. The final query image the is fed into the CMVD algorithm; d) 800 × 745 pixels. The current sketch and the image to subtract for the next sketch; e) 800 × 745 pixels. The Isolated object that resulted by subtraction of image 3a from image 3d; f) 100 × 100 pixels. The final query image is fed into the CMVD algorithm.*

Similarity Search in Content-Centric Networking

The CCN architecture as described in [4] and implemented in the CCNx project [10] permits retrieval of content, provided that the consumer knows all or at least a prefix of the name of the desired content object. Although this design has some serious advantages, it does not solve the problem of similar content searching, which is one of the most critical issues for future media Internet architectures. Our previous work in [5] is a first attempt to face this issue by proposing a search protocol as an extension to the CCN architecture. In the current work we use the aforementioned protocol to build a searchProxy daemon that works on the user space in the Linux system, as well as a gateway that works on the edge of the CCN network in order to interface it with other applications over TCP/IP networks.

The search-enabled CCN network as depicted in Fig. 1 has two basic components: the CCN gateway, which is responsible for the interconnection of the CCN with the TCP/IP network, and the CCN party, which is a node of the network that acts as a consumer and producer of data. The CCN search gateway computer should have the searchGateway application and a CCND [10] daemon that is the actual CCN router. For the CCN party the applications needed are the CCND router, the searchProxy application that implements the search protocol and conducts the searches for similar objects, and a file proxy application. The file proxy application is an implementation of a CCN repository available as a demo application with the CCNx project distribution [10].

The searchGateway and searchProxy subsystems are presented in the following two subsections.

The searchGateway

The CCN searchGateway is a Java application that runs on a Linux computer and interfaces the CCN with the TCP/IP network. The basic operation of the application is to receive search

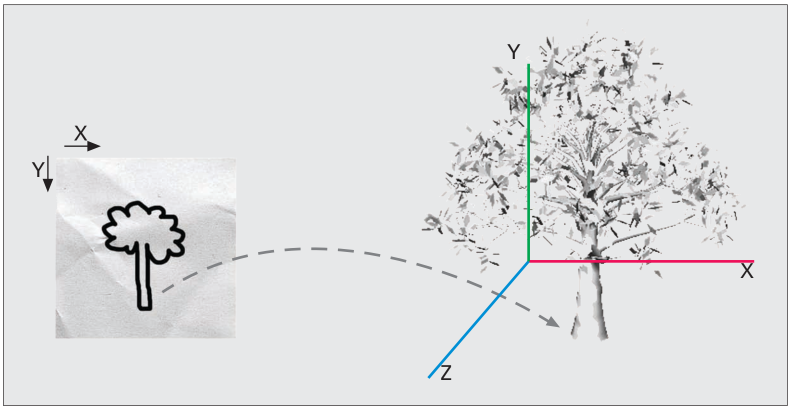

Figure 4. *Map X-Y coordinates from the sketch image to X-Z coordinates of the 3D world.*

requests from outside the CCN network (via TCP connections), form the Interest queries, and submit the search Interests to the search-enabled CCN network. If there are similar objects in the CCN network, the CCN searchGateway first receives and caches the similar content from the CCN parties, and then communicates with the client that started this search to send the requested content using FTP.

The gateway in idle time *listens* on a TCP socket for new clients from the IP side. When a new client requests a connection, a new thread is initiated that handles the connection. After the connection is established, a handshake procedure is followed to confirm that this is a valid client, and finally, the descriptor vectors from the query object are sent to the CCN gateway to form the search Interest. Figure 5 presents the messages exchanged by the CCN gateway and the sketchTo3D application for the initiation of a search session.

Upon reception of the descriptors vector from the sketchTo3D client, the searchGateway forms a search Interest, described in detail in [5]. In short, the search Interest name contains the descriptor vectors as well as the local name prefix in order for the CCN parties to refer to the CCN gateway that expressed the Interest.

While the main loop of the gateway waits for new search requests, the thread that expressed the Interest enters in a wait state, waiting for the CCN search-enabled parties to answer with similar multimedia objects. As a result of a successful search in a CCN party, a list of content names is sent to the gateway. The first record of the list refers to a file containing the ranking of the successfully retrieved content and the distance of each one from the query.

The CCN searchGateway waits for a predefined time window for responses and finally uses the ranked lists from the CCN parties to re-rank the available content. The re-ranking is based on the Euclidean distance (L2) of each content descriptor vector to the descriptor vectors of the query object. The result of this procedure is a new ranked list from which the top *K* most similar 3D models are retrieved and cached in the CCN searchGateway. Moreover, for every file that is cached in the temporary FTP directory of the gateway, a message is sent to the sketchTo3D application informing it that a result is available as well as the ranking of this result. This message is transmitted through the TCP socket that was initially established for the transmission of the query descriptors. As described above, the sketchTo3D application gets the resulting 3D model files using the FTP client.

The searchProxy Application

The searchProxy is also a Java application. Each search-enabled CCN party must have a searchProxy running in the background in order to support the CCN search protocol. A searchProxy instance is responsible for indexing the content that is available on the party's local repository and reply to search queries if similar content exists in its index.

For each 3D object in the local repository (file proxy) that has to be indexed, CMVD [9] descriptors (212 values for each view, 18 views in total) are extracted and saved on a local database. The searchProxy application uses a kd-tree indexing structure implementation in order to organize the records and perform fast exact searches or nearest neighbor searches. The search process is as follows.

First, a nearest neighbor search is performed to find the 10 most similar records in the database. Since the database consists of the descriptor vectors of the views of the 3D models, sometimes there are more than one views of the same 3D model that match the query and appear in the returned list. As a result, in a second step possible double records of 3D models are removed from the returned list. Third, the Euclidean distance is calculated between the query descriptors and each nearest neighbor in order to have the exact distances. Next, the 3D objects with distances greater than a threshold are also discarded from the results list. Based on the remaining objects and their distances, a ranked list file is created in the local repository containing information such as each object's name, distance from the query, and the rotation matrix of the *winning view* of the 3D object. Then a collection of objects is compiled with the first one being the name of the ranked list file.

After the compilation of the collection of 3D models that successfully passed the similarity matching process, a reply Interest is expressed to advertise the available results. When this Interest reaches the searchGateway, it is on the searchGateway's side to collect the desired content from the file proxy that serves these content objects as described in the previous subsection.

Experimental Setup

The experimental setup consisted of one Windows PC that hosted the sketchTo3D application and two Windows PCs that hosted in total four virtual machines running the Ubuntu Linux operating system in order to form the CCN network.

The first virtual box (VB1) worked as both a CCN search gateway and a CCN party. In other words, both searchGateway and searchProxy applications were running on the VB1 virtual machine. All the other virtual boxes (VB2, VB3, and VB4) played only the CCN party role of the network.

For the 3D models database we used the SHREC 2008 generic models track, which contains 1814 3D models of various objects (humans, vehicles, plants and flowers, etc.). The database was manually split into four overlapping parts, and each part was stored on a different virtual box of the CCN network in order to have different records on the different nodes of the network. However, we introduced a small overlap in order to test how duplicates would be handled from the re-ranking process of the gateway application.

Conclusions and Future Work

In this article we have presented a search and retrieval scheme that uses a single sketch to retrieve 3D models and compile a 3D environment by using the available multimedia content traveling in a content-centric network. By expanding the content-centric network to support multimedia similarity content search, we provide users the ability to retrieve multimedia content already available in the network without knowing where this content is stored or the name of each content in question. On the other hand, the sketch-based user interface is an intuitive UI that provides a greatly enhanced user experience for multimedia content search while helping users express in detail their thoughts and ideas.

For the future, as far as the user interface is concerned, we plan to rebuild it in a web application in order to have a wider group of testers and extend the user's actions available in the current version. Also, we are considering inserting real 3D video streams so as to create real-time on-the-fly 3D immersive environments from simple sketches.

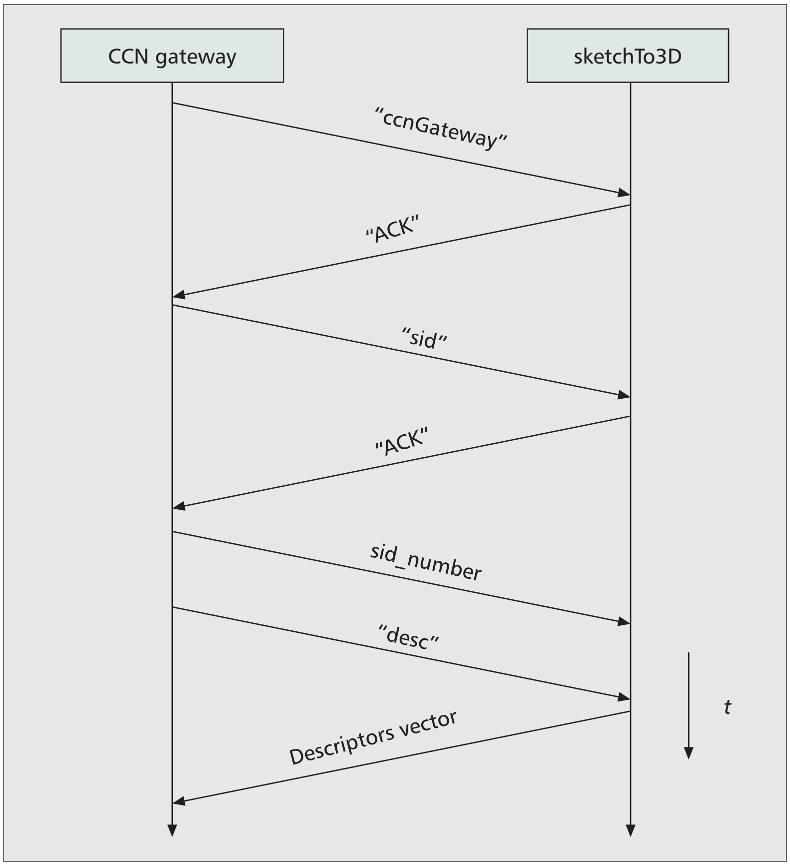

Figure 5. *Handshake messages between the CCN gateway and a sketchTo3D client submitting query descriptors.*

Acknowledgments

This work was supported by the EU FP7 project 3DLife, ICT-247688.

References

[1] N. Niebert *et al.*, "The Way 4WARD to the Creation of a Future Internet," *IEEE 19th PIMRC '08*, Sept. 15–18, 2008, pp. 1–5.
[2] G. Bouabene *et al.*, "The Autonomic Network Architecture (ANA)," *IEEE JSAC*, vol. 28, no. 1, Jan. 2010, pp. 4–14.
[3] T. Koponen *et al.*, "A Data-Oriented (and Beyond) Network Architecture," *Proc. ACM SIGCOMM '07*, Aug. 2007.
[4] V. Jacobson *et al.*, "Networking Named Content," *CoNext*, 2009.
[5] P.Daras *et al.*, "Similarity Content Search in Content Centric Networks" *ACM Multimedia*, 2010, Firenze, Italy.
[6] T. Chen *et al.*, "Sketch2Photo: Internet Image Montage," *ACM Trans. Graphics*, vol. 28, no. 5, Dec. 2009, pp. 1–10.
[7] M. Eitz *et al.*, "PhotoSketch: A Sketch Based Image Query and Compositing System," *ACM SIGGRAPH '09*, Aug. 3–7, 2009, New Orleans, LA.
[8] J. Pu, K. Lou, and K. Ramani, "A 2D Sketch-Based User Interface for 3D CAD Model Retrieval," *Comp. Aided Design App.*, vol. 2, no. 6, 2005, pp. 717–27.
[9] P. Daras and A. Axenopoulos, "A Compact Multi-View Descriptor for 3D Object Retrieval," *7th Int'l. Wksp. Content-Based Multimedia Indexing*, 2009, pp. 115–19.
[10] Project CCNx, accessed July 2010; http://www.ccnx.org/.

Biographies

THEODOROS SEMERTZIDIS received a Diploma degree in electrical and computer engineering from Democritus University of Thrace (2004) and an M.Sc. degree in advanced computer and communication systems from Aristotle University of Thessaloniki, Greece (2009), where he is now a Ph.D. candidate. He has worked for the Informatics and Telematics Institute as a research associate since 2006. His research interests include distributed systems, and multimedia search and retrieval.

PETROS DARAS [M'07] (daras@iti.gr) is a senior researcher at the Informatics and Telematics Institute. He received a Diploma degree in electrical and computer engineering, an M.Sc. degree in medical informatics, and a Ph.D. degree in electrical and computer engineering from Aristotle University of Thessaloniki in 1999, 2002, and 2005, respectively. His main research interests include computer vision, search and retrieval of 3D objects, and medical informatics.

PAUL MOORE is a graduate in computer business systems of Ryerson University, Toronto, Canada, and also holds a degree in economics from the University of Toronto. He has more than 20 years of experience in IT systems, including six years as technical director or coordinator of different European projects. He is head of the Media unit in Atos Research & Innovation, and is the representative for Atos Origin on the Steering Committee of NEM.

LAMBROS MAKRIS is a research associate at the Informatics and Telematics Institute, Greece. He received his Diploma and Ph.D. in electrical engineering from Aristotle University of Thessaloniki in 1994 and 2007, respectively. His research interests include applications of local area and wide area networks, distributed information systems, databases, electronic commerce, data security, and encryption.

MICHAEL GERASSIMOS STRINTZIS [M'70, SM'80, F'04] received a Diploma degree in electrical engineering from the National Technical University of Athens, Greece, in 1967, and M.A. and Ph.D. degrees in electrical engineering from Princeton University, New Jersey, in 1969 and 1970, respectively. He is a professor of electrical and computer engineering at the University of Thessaloniki. He has served as Associate Editor for *IEEE Transactions on Circuits and Systems for Video Technology* since 1999. In 1984 he was awarded one of the Centennial Medals of the IEEE.