An Optimization Approach to Group Coupling in Heterogeneous Collaborative Systems

Carlos D. Correa and Ivan Marsic

Rutgers University
ECE Department and the CAIP Center
96 Frelinghuysen Road
Piscataway, New Jersey 08854-8088 U.S.A.
+1 732 445 0542

{cdcorrea, marsic}@caip.rutgers.edu

ABSTRACT

Recent proliferation of computing devices has brought attention to *heterogeneous* collaborative systems, where key challenges arise from the resource limitations and disparities. Sharing data across disparate devices makes it necessary to employ mechanisms for adapting the original data and presenting it to the user in the best possible way. However, this could represent a major problem for effective collaboration, since users may find it difficult to reach consensus with everyone working with individually tailored data. This paper presents a novel approach to controlling the coupling of heterogeneous collaborative systems by combining concepts from complex systems and data adaptation techniques. The key idea is that data must be adapted to each individual's preferences and resource capabilities. To support and promote collaboration this adaptation must be *interdependent*, and adaptation performed by one individual should influence the adaptation of the others. These *influences* are defined according to the user's roles and collaboration requirements. We model the problem as a distributed optimization problem, so that the most useful data— both for the individual and the group as a whole—is scheduled for each user, while satisfying their preferences, their resource limitations, and their mutual influences. We show how this approach can be applied in a collaborative 3D design application and how it can be extended to other applications.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces; H.2.5 [**Database Management**]: Heterogeneous Databases—*Data translation*; D.2.11 [**Software Engineering**]: Software Architectures

General Terms

Algorithms, Performance, Design, Experimentation

Keywords

Coupling, 3D content adaptation, heterogeneous systems, CSCW

1. INTRODUCTION

With the increasing popularity of data networks and mobile devices, there has been a demand for technologies that support collaboration of geographically separated knowledge workers. Here we address a major bottleneck to the rapid growth of collaborative applications—that of heterogeneous computing platforms with unequal computing and communication resources.

When users collaborate via dissimilar devices, system resources are often constrained, so it may not be possible to transmit, store, and visualize a dataset in an unaltered form. For this reason, applications need to adapt the data to match the available resources by supporting different grades of representation or fidelity, or different modalities of representation. This is a critical requirement: the variety of device capabilities requires a corresponding variety of representational types and fidelities, each having different resource demands. For instance, it is common in graphical applications to obtain different representations through different rendering techniques, (such as wireframe, illustrative, and photorealistic [11]), through simplification, or through filtering. An example is shown in Figure 1. Similarly, collaborative applications with audio and video can be adapted through different media encoders, compression levels and bit rates. This creates an economy of representations, within which the system can make choices among alternatives and allocate resources to achieve the individual and group objectives.

A key issue raised by such adaptive applications is whether the resulting heterogeneity allows meaningful collaboration. In some applications, such as navigation of a shared 3D world, this may not be a problem since there is no critical decision making involved and the main goal is to enable participation with resource-limited devices. However, for other kinds of applications, e.g., in product design, it may be necessary to provide similar or identical views for some or all participants, since reaching a consensus or assuming accountability for the decisions may critically depend on having as similar a representation across the participants as possible. This preservation of similarity is commonly known as *coupling*.

This paper presents an approach for enabling coupling in collaborative systems through optimization. This optimization seeks to satisfy both the users' preferences and system capabilities, and the collaborative constraints. In other words, the system is able to strike an optimum balance between selfishness and sympathy of the group members. To achieve this, we model

collaborative groups as a network of adaptation processes, which are responsible for solving the optimization problem. The group requirements for effective collaboration are modeled as *influence links*, which define the way the data adaptation of a given participant limits the options of data adaptation for the others. This distributed optimization is a hard problem since it is a variation of the well-known knapsack problem, which is NP-complete [27], and, in general it may not be possible to solve optimally in a strictly decentralized setting.

Our approach offers a model and framework that can be applied towards developing tools for collaborative design, development and visualization by open source communities and industry. In this paper we present a particular application of our model to collaborative 3D design, and we show how these ideas can be extended to other types of applications.

The rest of the paper is organized as follows: Section 2 describes previous work related to our model. The optimization framework and the ideas of influence links are described in Section 3. Section 4 presents some application scenarios. Section 5 shows an evaluation of our model to validate the model as an effective mechanism for handling coupling in heterogeneous collaborative systems. Finally, we present conclusions of our work.

2. RELATED WORK

Our approach is related to three main research fields: (1) *data adaptation*, (2) *multi-resolution representations*, and (3) *data and view coupling*, all in the context of heterogeneous collaboration.

Data adaptation can be applied at different levels of abstraction, but usually requires knowledge of the application semantics in order to be more effective, such as in systems like Odyssey [29] or in [13]. The core of these solutions lies in adaptively compressing data so that the storage, transmission, and visualization costs are reduced, without a significant impact on its utility to the user. The range of applications where data adaptation has been exploited is wide: from transcoding the Web content [3][12][24], through text summarization [25][26], to image and video compression [28] and adaptive 3D applications [14]. There has been a recent effort to support collaboration over mobile devices through data adaptation [7][31]. Ref. [7] works with multi-representations, but concerns itself mostly with consistency issues due to concurrent updates. Ref. [31] uses optimization to allocate resources so to maximize the individual user benefit, but ignores the group interests.

The role of data adaptation in networked virtual environments has been limited so far to the ability to transmit and store large 3D datasets on a variety of platforms [6][18]. However, the challenges of interactive collaboration in the presence of heterogeneous 3D representations have not been explored in depth. A recent approach by Cera *et al*. [4], addresses the problem of collaboration over CAD models among users with different roles and security constraints for data access. They use multi-resolution models to provide transparent data adaptation to the users according to their respective access privileges.

The use of multiple representations in collaborative systems imposes additional challenges in providing similar views and/or representations of the shared data for the different users. Correa and Marsic [5], and Li and Rui [23], address the problem when interoperating heterogeneous systems, particularly shared editors.
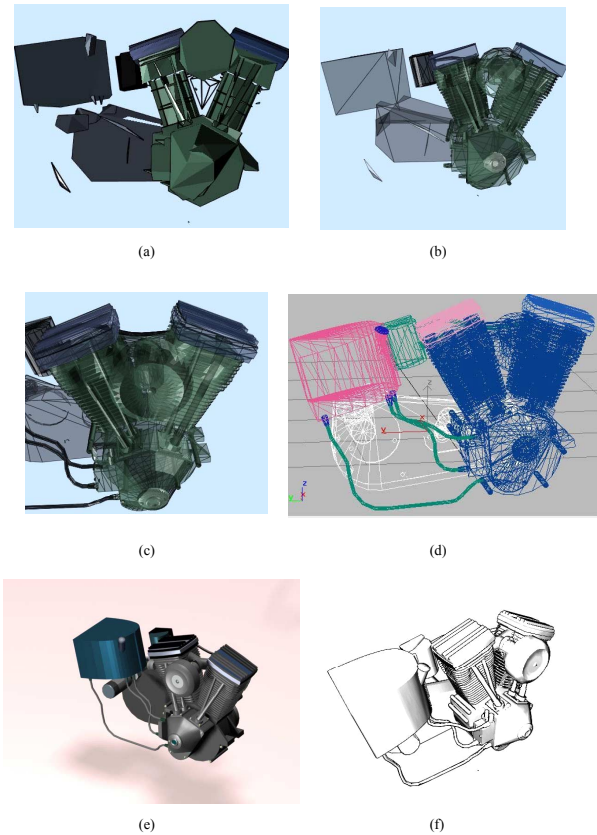

Figure 1. Multiple representations in 3D design applications (a) and (b) are obtained by simplification, while (c)-(e) use different rendering styles: (c) shaded (d) wireframe (e) photorealistic (f) illustrative.

Coupling has been the focus of multiple research approaches in the literature, e.g., DistEdit [22], Corona [16], Jasmine [33]. The tradeoff between group and individual satisfaction has also been studied in [8][15], from the perspective of user interface and group awareness but they do not consider data adaptation to system resources. In this paper, we address the problem of group coupling at the level of data representation, and the techniques and widgets at the user level can be incorporated on top of our approach. Our prior work [5] presents a framework to interoperate heterogeneous collaborative systems, based on the adaptation of the underlying data structure of the shared dataset. The data structure, represented as a graph, included both the structural and semantic relationships between the different shared elements. However, this framework only supports individual adaptation of the data and does not provide a mechanism for ensuring coupling among the different users. In this paper, we enable users to specify their preference towards individual vs. group satisfaction of collaborative policies. Our model optimizes the data adaptation to provide the users the best possible representation of the shared dataset while satisfying their group preferences, which can be set up with a small number of parameters. In the next section, we describe the optimization framework for heterogeneous collaborative systems.

3. OPTIMIZATION FRAMEWORK

A collaborative session is a group of users interacting with a set of tools over a shared dataset. Users may have different *representation* of the data, due to limitations in resources, such as computing cycles or network bandwidth, or due to the user's preferences. To allow adaptation, we assume that the dataset can be represented in multiple forms.

3.1 Multi-representation of Shared Data

Multi-representation of shared data makes possible collaborative work from a variety of devices. Applications can be extended to support multi-representation data by adding an extra dimension to their elements. According to this paradigm, each element or object *j* in a shared environment can have up to *L* distinct representations. Each of these representations has an increasing *cost r*, and a benefit *b*. The cost can be defined as CPU, memory or bandwidth requirements. For instance, in 3D environments, the number of polygons used to represent an object is a good cost metric, since it represents both the memory requirements, as its rendering cost. Benefit metrics represent the quality of a given representation for the user. Although this may seem subjective, quantitative measures, such as image similarity, are proven to be good metrics of benefit [14].

Examples of multiple representations for 3D environments can be obtained from simplification, different rendering styles, or from model abstraction, as depicted in Fig. 1. The corresponding resource requirements and rendering speeds are shown in Table 1. For images and video, multiple representations can be obtained using different compression levels, resolutions, or color palettes.

Table 1. Multi-representation characteristics for Figure 1.

Represent.	Triangles	Rendering	Speed
Fig. 1(a)	3674	Flat shading + Wireframe	High
Fig. 1(b)	12546	Flat shading + Wireframe	High
Fig. 1(c)	16624	Flat shading + Wireframe	High
Fig. 1(d)	16624	Wireframe	Higher
Fig. 1(e)	16624	Phong shading + Shadows	Low
Fig. 1(f)	16624	Illustrative rendering	Medium

3.2 Individual/Local Optimization

The first step of our optimization framework is to ensure that individuals have the best representation of the shared data that satisfies the resource constraints and their preferences. For this purpose, we assume that each participant *i* has assigned the value of benefit *automatically* as in [14], $b_{i,j,k}$, and cost, $r_{i,j,k}$, to each representation *k* of every object *j* in the shared environment.

We solve the problem of local resource scheduling as an optimization problem, similar to the well-known Knapsack Problem [27]. Its solution is a set of binary values that tell us whether or not a given representation *k* of an object *j* is selected for a given user.

More formally, let us define a set of *N* users sharing *M* objects each of which has *L* representations of different fidelities. We also define a limit of resources R_i available on the device of user *i*. Let

$$x_{i,j,k} = \begin{cases} 1 & \text{if representation } k \text{ for object } j \text{ is selected for user } i \\ 0 & \text{otherwise} \end{cases}$$

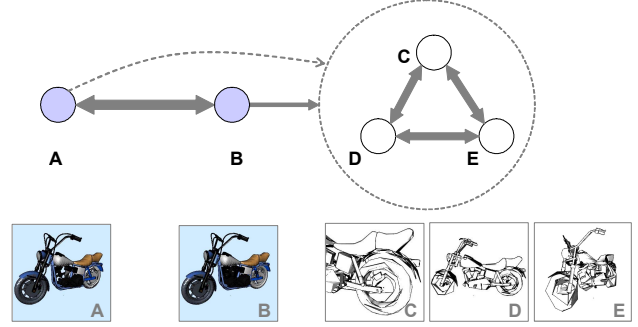Then, the optimal scheduler must solve:

Figure 2. Five users sharing a 3D model and their respective views. Arrows indicate capability to influence another user's view. The arrow's thickness indicates the influence strength.

$$\max \left\{ \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{L} b_{i,j,k} \cdot x_{i,j,k} \right\} \quad (1)$$

$$\sum_{j=1}^{M} \sum_{k=1}^{L} r_{i,j,k} \cdot x_{i,j,k} \leq R_i \qquad 1 \leq i \leq N \quad (2)$$

$$\sum_{k=1}^{L} x_{i,j,k} = 1 \qquad 1 \leq i \leq N, \quad 1 \leq j \leq M \quad (3)$$

The objective function we need to maximize is given by (1), which is the overall sum of *benefits*. It is subject to the *resource limitations* of each user's computing device, (2), and the *unique representation constraint*, (3), which says that there must be exactly one representation selected per object. Note that solving this problem for all users is the same as solving it for each one independently. Thus, this is referred to as *local optimization*.

Local optimization is an effective way to provide the users with the best representation of the shared data that satisfies their constraints. This type of optimization has been used widely in works such as [31]. However, this method is not sufficient for effective collaboration, since it ignores the group interests and focuses only on individuals.

3.3 Coupling/Influence Links

We define *coupling* as the degree of similarity between the dataset representations for different users in a collaboration group. Complete coupling is achieved when the different users have the same representations for all the shared elements, while no coupling is achieved when the representations of different users are independent. Notice that this definition does not concern their *viewpoints* but only the *fidelity* of their representations. We will discuss viewpoint coupling in Section 4.1.1 below.

Furthermore, coupling can be specified differently for users with different roles. This coupling is directly related to the *influence* of one user upon another. To understand this concept of influence, let us consider the scenario depicted in Figure 2: Five users share the same 3D object using a variety of platforms. User *B* has a role of coordinator and oversees the work of three peers *C*, *D* and *E*. These peers share the same interests and responsibilities for the task, so it is expected that they would have identical or similar representations, but since their computing capabilities may not be the same, they may have some differences. Thus, there is a strong coupling among them. As depicted in Figure 2 (bottom row),

these three peers have an illustrative rendering type of representation, composed of simplified versions of the objects seen by user *B*. Similarly, users *A* and *B* have a strong coupling, and their display views are almost identical. We can say that users *A* and B have a strong symmetric influence over each other, and ditto for users *C*, *D* and *E*. Conversely, user *B* has a medium influence on *C*, *D*, *E* and it is asymmetric so *B* is not hampered by the lack of resources available to *C*, *D*, and *E*. Similarly *A* has a weak (and asymmetric) indirect influence on *C*, *D*, and *E* (indicated by the dashed arrow). From this scenario, we identify two types of mechanisms by which one user influences another:

- *Peer influence links*: These links are *symmetric*, and they are used between individuals with the same role. Here, each individual will strive to have the same representation for the same object. Thus, each individual's optimal scheduler influences the other peers to make similar, or even identical, decisions about the scheduling. In other words, the *difference* between the object representations across the group tends to be minimal.

- *Coordinator influence links*: These are always *asymmetric* links, where one user plays a more *important* role than the other. We say that they are related through a coordinator-coordinated relationship. In such a case, it is safe to say that the selection of a representation on the coordinator's site influences the coordinated user to be similar or identical, but selecting a representation in the coordinated user does not necessarily influence the coordinator.

This distinction of influence links is used in this paper to refer to *symmetric* and *asymmetric* influences. It is important to distinguish this division from user roles. Roles in collaboration are not restricted to peer or coordinator-worker, but include other types such as information recorder, observer, moderator, among others. We believe that the different types of roles imply different degrees of influence among users. For instance, a user with a role of *information recorder* has little, if any, influence onto the others, while his (her) actions are influenced by all users.

Note that in a real collaboration scenario these influences may be formed dynamically, rather than being defined *a priori*. For instance, one user may be the coordinator of a subgroup of two peers. It is sensible to think that a coordinator influence link is formed when the coordinator is overseeing the work of the others. However, when this user needs to reach a consensus with the coordinated two, this requires common grounding and it is important to have similar representations, even if that compromises the local utility of some individual. In such case, the peer links are formed among all three users.

We emphasize that these influences are *not generally* in force for the entire collaborative session. Rather, they come into effect *only at critical junctures*, when joint decisions are being made.

3.4 Group/Global Optimization

As noted by Klein *et al*. [21], the optimal state of a complex system can be obtained by maximizing the degree of satisfaction of the influences between their elements. In our case, the optimal state refers to the particular representations of shared data selected for each user at a given time. The influence satisfaction can be introduced into the optimization in two ways: influences as a
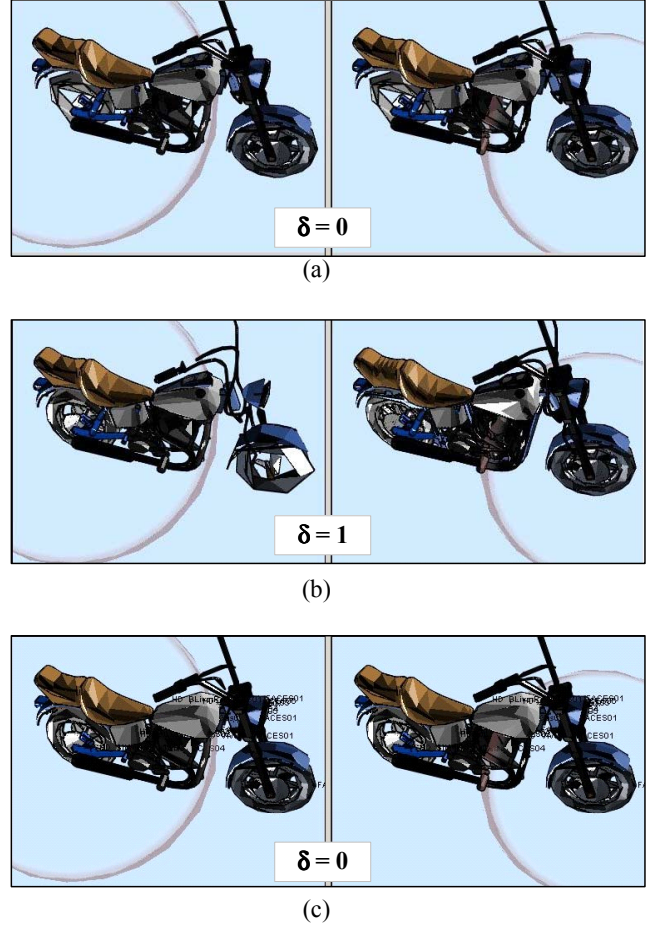

Figure 3. Global optimization with delta satisfaction (a) $\delta = 0$; (b) $\delta = 1$; (c) relaxation of the completeness requirement, $\delta = 0$. The user seeing the left view has lesser resources. The views in (c) are exactly the same. Text labels indicate omitted elements.

constraint, or influences as part of the objective function, as described below.

3.4.1 Influences as a Constraint

In this case, a new constraint is added to the problem. Let $\Im$ denote the subset of all hosts in a collaborative configuration that are connected by influence links. Following the notation defined in Eqs. (1)–(3), let a parameter $\delta_{i_1,i_2} \in [0,L]$ denote the tolerable difference between the object representations of users i_1, $i_2 \in \Im$. If i_1 and i_2 exert no mutual influence, then $\delta_{i_1,i_2} = L$. If there is a *coordinator influence link* from a coordinator i_1 to a coordinated user i_2, then $\delta_{i_1,i_2} < L$. If there is a *peer influence link* between i_1 and i_2, then $\delta_{i_1,i_2} = \delta_{i_2,i_1} < L$. The greatest influence is when $\delta_{i_1,i_2} = 0$, where both users must have identical views.

The δ parameters can be specified by the group, the application developer, or some coordination entity, and can be changed dynamically. We assume that the different representations, indexed by subscript k, are ordered in ascending degree of fidelity. Then, the constraint

$$\sum_{k=1}^{L}\left(k\cdot\left(x_{i_1,j,k}-x_{i_2,j,k}\right)\right)\le\delta_{i_1,i_2}\qquad i_1,i_2\in\Im,\quad i_1\ne i_2,\quad 1\le j\le M \qquad (4)$$

says that for all pairs of hosts i_1 and i_2 that exert mutual influence, the fidelity-wise difference of representations for user i_1 cannot exceed that of user i_2 by more than δ_{i_1,i_2}. We refer to the problem stated by Eqs. (1)–(4) as *global optimization with delta satisfaction*. Note that the absence of absolute value in Eq. (4) represents an *asymmetric* influence. For a symmetric influence, we would have two constraints of the type of Eq. (4), with switched indices δ_{i_1,i_2} and δ_{i_2,i_1}, which is equivalent to a single constraint with the absolute value in the summation in Eq. (4).

Notice also that δ is inversely related to the strength of the coupling link. For instance, when $\delta = 0$, representations for users i_1 and i_2 are identical, i.e., the coupling is maximal. When $\delta = 1$, the difference between representations must not exceed 1. When $\delta = L$, i.e., the maximum number of different representations per object, this constraint is always satisfied, so the global utility is just the sum of local utilities, i.e., the coupling is minimal.

Figure 3 shows an example of this mechanism. Three different stages of a collaborative session are shown. Each row shows the view from two different users with different resource limitations. The images on the left correspond to *user1* with approximately a third of resources available to *user2* who sees the images on the right. For illustration purposes only, it is assumed that the users share the viewpoint. In an actual collaborative session, it may be beneficial for users to have independent viewpoints that may occasionally be coupled *on demand* for common grounding. The transparent spheres in the images correspond to the region of interest (focus of attention) of each user. For instance, *user1* is interested in the back wheel of the motorcycle, while *user2* is interested in the front one. Figures 3(a) and 3(b) illustrate the results of delta satisfaction. In this case, we see that setting $\delta = 0$ results in the highest coupling possible, since the constraint forces the data adaptation system to provide identical views for both users. However, we see that this penalizes *user2* by choosing low fidelity for some elements although resources are left unused (see, e.g., the back wheel). A way to avoid this is to increase the constraint to $\delta = 1$. In this case, each user gets a high fidelity element in their own region of interest (*user1* gets to see the back wheel in high detail), but the coupling is low. This constitutes the main disadvantage of delta satisfaction. A small change in the coupling parameters, i.e., delta, results in drastic view changes for both users. In a collaborative application, this is distracting and results in very low resource efficiency.

This mechanism implies that coupling requirements are hard constraints that must be satisfied. For some applications, this may be too rigid to provide access for users with limited resources. For this reason, it may be more appropriate to introduce influences as soft constrains, as described below.

3.4.2 Constraint Relaxation
A mechanism for remedying the brittleness of the influence constraints is to relax some of the other optimization constraints.
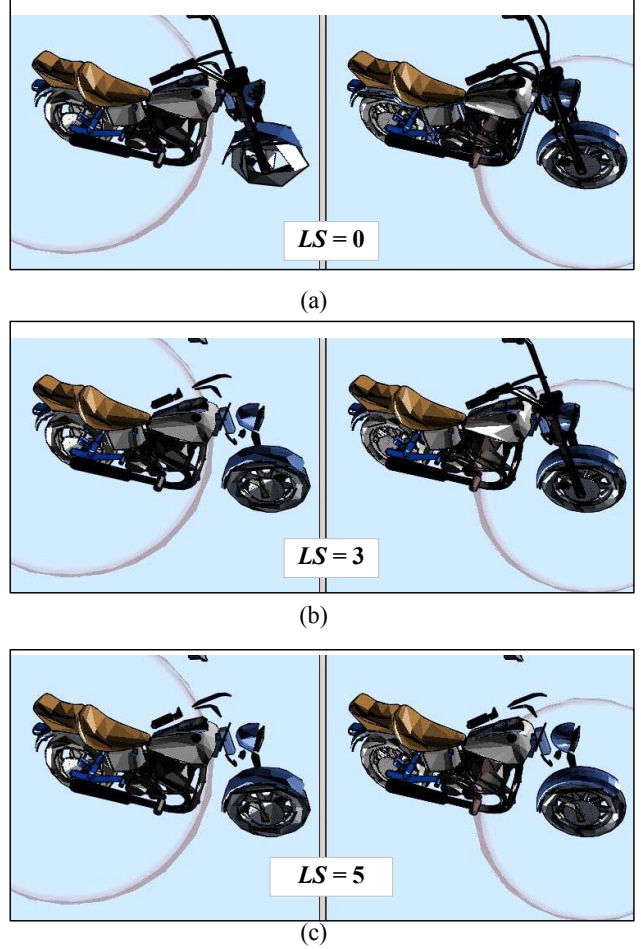

(a)

(b)

(c)

Figure 4. Global optimization delta minimization for (a) Link strength = 0 (low coupling); (b) Link strength = 3 (medium coupling); (c) Link strength = 5 (higher coupling). The user seeing the left view has lesser resources. Going from (a) through (b) to (c), the fidelity of the left view is relocated from the handles to the front wheel (focus of attention of the right user), while the fidelity of the right user decreases, starting with the handles which are of no interest to either user.

So far, we have required from the optimization approach to provide complete, one-to-one versions of the shared dataset. This is expressed by Eq. (3), which enforces exactly one representation per object. We can relax this constraint to open up the possibilities for data adaptation when we introduce influences as constraint, Eq. (4).

Instead of exactly one representation, we modify the constraint to allow *at most* one representation per object, as shown in Eq. (3b).

$$\sum_{k=1}^{L}x_{i,j,k}\le1\qquad 1\le i\le N,\quad 1\le j\le M \qquad (3b)$$

$$\max\left\{\sum_{i=1}^{N} w_0^i \cdot \sum_{j=1}^{M}\sum_{k=1}^{L} b_{i,j,k} \cdot x_{i,j,k} - \sum_{i_1,i_2 \in \Im} w_1^{i_1,i_2} \cdot \sum_{j=1}^{J}\left|\sum_{k=1}^{L}(b_{i_1,j,k}+b_{i_2,j,k})\cdot(x_{i_1,j,k}-x_{i_2,j,k})\right|\right\} \tag{5}$$

$$\max\left\{\sum_{i=1}^{N} w_0^i \cdot \sum_{j=1}^{M}\sum_{k=1}^{L} b_{i,j,k}^2 \cdot x_{i,j,k} - \sum_{i_1,i_2 \in \Im} w_1^{i_1,i_2} \cdot \sum_{j=1}^{J}\left(\sum_{k=1}^{L}(b_{i_1,j,k}+b_{i_2,j,k})\cdot(x_{i_1,j,k}-x_{i_2,j,k})\right)^2\right\} \tag{6}$$

This constraint allows the data adaptation process to omit elements of low importance and use their resources to select higher fidelity versions of more important elements. Figure 3(c) shows an example of this relaxation for $\delta_{i_1,i_2} = 0$, i.e., the highest possible coupling. As seen, the result is a more detailed representation of the most important objects (those that are attended by either user), while unimportant (unattended) elements are omitted. We show text labels for the omitted objects (centered on the omitted object's center of mass). The text labels provide context and introduce no significant rendering cost. As it can be seen, both users get a better representation and still high coupling is enforced. Compared with Figure 3(a), the overall detail is higher for both users.

3.4.3 Influences as Part of the Objective Function

In contrast to constraint satisfaction, one may want to obtain a solution that balances the benefit of each individual with the benefit of the group. Based on the definitions of influence links, maximizing the benefit of the group can be defined as minimizing the differences between the users' representations. This can be thought of as a *soft constraint* on the coupling parameters, unlike the above approach, which enforces hard constraints. Constraint (4) must always be satisfied, whereas embedding influences in the objective function does not guarantee constraint satisfaction—hence its "soft" nature.

In the optimization model, we can account for the dissimilarity of users' representations as a penalty function to the sum of local utility (benefits). This penalty function can be incorporated in the optimization problem as a sum that is subtracted from the overall benefit. This way, the higher the differences between the representations, the lower the benefit is. The modified objective function becomes either Eq. (5) or (6). In (5) the penalty function is defined as the sum of absolute differences (SAD) between the fidelities of all objects, weighted by the sum of their benefits. Conversely, in (6) the penalty function is defined as the sum of squared differences (SSD). Note that in (6), benefit values in the first term of the sum are squared in order to maintain the same scale as in the second term.

We introduce two weighting parameters w_0 and w_1 to allow the application and users to control the relative importance of the preferences of the group vs. the preferences of the individuals. Weight w_0 is assigned individually to all users i, and w_1 is assigned only to the pairs of users i_1 and i_2 that exert mutual influence. When $w_0 > w_1$, more importance is given to preferences of the individuals, whereas when $w_0 < w_1$, more importance is given to the preferences of the group. Therefore, the ratio w_1/w_0 can be used to measure the strength of the influence links.

We call the problems defined by the equations (5), (2), and (3), or (6), (2), and (3) as *global optimization with delta minimization*.

The penalization function is applied to pairs of clients. This means that w_1 does not need to be identical across all the pairs of clients, which enables us to accommodate the different types of influence links defined above. Further, the manipulation of the optimization parameters enables the incorporation of collaboration aspects, including user autonomy, shared views (e.g., by setting the weight values to enforce homogeneous representation fidelity), different access rights (by setting different weights for each user) and global vs. local views (also known as detail vs. context tradeoff [32], obtained by manipulating the benefit values of the different objects).

Figure 4 illustrates the results for delta optimization with different link strengths, on the same scenario as in Figure 3. Link strength, LS, is determined as the ratio between the weight used for delta minimization and the weight for benefit maximization.

$$LS_{i_1,i_2} = \frac{w_1}{w_0} \tag{7}$$

Figure 4(a) shows the view for the two users for a link strength LS = 0, i.e., no coupling. Figure 4(b) and Figure 4(c) show their views for LS = 3 and LS = 5, respectively. It can be seen that this results in better balance of resources and preference than for influence satisfaction. Further, by changing the parameters, i.e., LS, users can smoothly explore the shared dataset from no-coupling to high-coupling settings. This is further described below in the evaluation section, where we quantify coupling and resources as the link strength parameter is varied.

3.5 Collaborative Policies

The group policies include the choices about optimization strategies and particular coupling characteristics. For example, one choice is between global optimization with delta satisfaction vs. with delta minimization. In the former case, the tunable parameters are δ_{i_1,i_2} where $i_1, i_2 \in \Im$, $i_1 \neq i_2$, are any pair of users that exert mutual influence. In the latter case, the tunable parameters are the link strengths, LS_{i_1,i_2} in Eq. (7), between the different users. Currently the influence links are defined only pairwise for a set of individual hosts. A possible topic for future research is to extend this to hierarchical influences, such as coordinator-to-group or group-to-group.

4. ENVISIONED APPLICATION SCENARIOS

Some application scenarios of our approach are as follows.

4.1 Collaborative Editing

The main application we intend this work for is collaborative 3D editing, but the ideas presented here can be extended to other applications, such as collaborative multimedia authoring tools, collaborative design, and mobile video games. The main requirement for applying our approach is to have a mechanism to generate multiple representations of individual elements of a structured dataset. In some cases, this is already provided by the

application, as it is common in many multimedia authoring tools. This usually is provided through different compression level, resolution, or sampling frequency of the media. For other cases, many applications provide an API that can be extended to enable multi-representation data and adaptation.

The users will typically not create the 3D models manually; rather, these will be retrieved over the network and imported into the collaborative application. The models could be CAD models obtained from a database, volumetric models obtained from a collection of images, or 3D images reconstructed from real-time sensors. Prior to retrieving the models, the collaborative application must run the optimization algorithm presented above, to determine the optimal representation selections $x_{i,j,k}$ to best fit the individual resources while complying with the group policies. Since the models could be retrieved from several different sources, it would not be feasible to run the centralized optimization algorithm at a model source; hence, developing a distributed optimization algorithm is part of future work.

The collaboration process would mostly consist of selecting the 3D models suited for the task at hand, retrieving them, and assembling into a final composition which solves the problem. As users join the collaborative session, influence links are formed according to their roles, their tasks and their immediate goals. For example, two CAD designers might join with a symmetric influence link so to have similar views. An observer can join with little or no influence over these two, but strongly influenced by what they do. As a result, the features important for the two designers will be well represented for the observer. A coordinator might also join intermittently to express important concerns and provide assistance. In this case, the coordinator forms an influence link of variable strength, which is set high when the coordinator actively participates and low when s/he passively observes.

4.1.1 Viewpoint Coupling

An important aspect of collaborative work is the management of the users' views. In 3D environments, it is common to represent the user viewpoint in the same data structure of the shared objects, e.g., as in VRML [35]. If we apply optimization to the viewpoint elements, it is possible to obtain some kind of *view coupling* in collaborative 3D. This idea has been exploited before in 2D shared editors and whiteboards, as in [8][15] and [30]. Tight view coupling is also known as WYSIWIS (What You See Is What I See), and has been a widely used paradigm in collaboration [34]. It has, however, been rarely employed in 3D environments. A notable exception is the work in [36], where different levels of view coupling are obtained via constrained navigation.

In our approach, viewpoint coupling refers to the distance between the viewing points of the different users. When two viewpoints are tightly coupled, the users are seeing exactly the same scene from the same perspective. When one user changes the viewpoint, that of the other user is changed in sync. In contrast, uncoupled viewpoints mean that the users can see a different view of the scene and the changes in one's do not effect changes in the other's. Between these two extremes, we can define a spectrum of coupling levels, based on the link strength, defined above. The higher the strength, the nearer their viewpoints are.

We can implement this model by *virtual springs* attached to the observer's location and the center of his/her region of interest. Similarly to the optimization above, these springs can be incorporated as hard or soft constraints. Since there are no multiple representations of viewpoint "objects," instead of satisfying constraints or minimizing the difference of the representations, we satisfy or minimize the spatial distance between the viewpoint locations. If this process is repeated whenever a user changes the viewpoint, they behave as if they were connected by virtual springs.

4.2 Shared Augmented and Mixed Reality

AR and MR applications [1][9][10] are becoming popular with the availability of commoditized 3D graphics hardware.

In these applications, the goal is to augment the view of the real world, usually obtained from a camera worn by the user, by overlaying digital information that may be too cumbersome to provide by other means. For instance, names of buildings may be attached to the view of a complex urban area, thus providing the means to locate a place of interest. For this reason, AR interfaces suffer from an important tradeoff between *augmentation* and *occlusion*, as identified in [17][20]. Overlaying too many virtual objects onto the real world input results in cluttering the view and occluding important objects, and becomes counter-productive. Conversely, overcautious avoidance of object occlusion might diminish the potential value of augmentation. For this reason, multiple representations are needed in augmented reality. Choices for representations are: wireframe, transparent, or solid objects, which have an increasing occlusion cost when overlaid over the actual objects, or arrows and textual labels, which have the benefit of providing useful information to the user without occluding large parts of the actual objects.

The above framework enables the creation of adaptive AR user interfaces that select appropriate representation for each object that is being augmented, so to balance the occlusion cost vs. the user benefit in the context of the task. For example, objects which are closer to the user and are vital to the task could be augmented using a transparent 3D model. Other less important objects could be represented using wireframe rendering, while all other objects could be augmented using labels, so as to provide peripheral awareness to the user. This adaptation also accounts for collaborative users, whether collocated or geographically dispersed. Users not only have the benefit of multiple representations to better perform their task, but also the benefit of sharing the views with others, and the possibility to gather more information about the real environment through collaboration.

5. IMPLEMENTATION ISSUES

5.1 Accuracy of Cost and Benefit Models

The first challenge is the acquisition of the cost $r_{i,j,k}$ and benefit $b_{i,j,k}$ parameters of the shared objects. Ideally, one would attempt to obtain accurate values of costs and benefits to best employ the framework. The underlying assumption is that the benefit in mathematical sense is equivalent to the user satisfaction (the perceived benefit). Some benefit and cost metrics can be defined a priori. In the context of 3D collaboration these include: polygon count and static memory requirements for the costs, and similarity metric (obtained from the mesh simplification algorithm) and visual size for the benefits [14]. Dynamic metrics, such as rendering cost (associated with window size in 3D editors), CPU cycles, pixel resolution and user's viewpoint, which must be

monitored in real-time, are difficult to obtain. This generally requires a monitoring module that continuously feeds the optimization algorithm.

An alternative is to work with approximate information and let the user(s) explore the framework until they find a workable solution. The search space spanned by the optimization problem can be explored by users manipulating the benefit and cost metrics and observing the impact of their choices. We know that the approximate values are within some distance from the actual values. The optimum solution found by the framework given the approximate cost/benefit model will be in the vicinity of the true optimum. Hence, it is meaningful to experiment and find a workable approximate model, without ever needing the accurate cost/benefit model. For this, a set of user interface widgets must be provided to quickly assign benefit/cost metrics automatically, and to asses the utility of those metrics. Since the framework is well-behaving, this exploration will eventually lead to the models that give acceptably optimal solutions from the user's perspective.

5.2 Maintaining the Influence Links

Another implementation aspect is the maintenance of the influence links. As mentioned above, influence links cannot always be derived a priori. This is because users' roles change during a collaborative session, and because users might join and leave frequently. An option is to implement a user awareness module that interacts with our framework. The application developer defines the rules to derive the influence links from the user awareness information. An example is the definition of roles through tables or XML files [2]. A set of rules can be defined to map roles into influence links. These rules derive the influence type (symmetric vs. asymmetric) and the strength.

5.3 Performance Issues

An important issue is the performance of the optimization algorithm. In the experiments described below, we used general purpose, commercial optimization software CPLEX by ILOG (http://www.ilog.com/products/cplex). General purpose solutions are usually based on well known algorithms such as branch-and-bound [19]. They are usually computationally expensive, and sometimes cannot be bounded in time for obtaining a provably optimal solution. Generic purpose solvers work for a wide number of constraints and objective functions but may not be fast enough for real-time optimization. For the kinds of applications that require real-time adaptation, it is better to implement specialized algorithms, such as those described in [27]. As pointed above, in a collaborative setting, it is also necessary to use distributed solvers, such as the optimization algorithms described in [37], rather than a centralized approach.

6. EVALUATION

Our main goal in the evaluation is to determine how effective the above model is in controlling the tradeoff between individual and group satisfaction. In particular, we want to know for different objective functions how well they behave when varying the control parameters. We prefer the objective functions that are:

- *Smooth*, such that small changes in control parameters do not result in dramatic changes in the resulting representation;

(a)   (b)

Figure 5. Two example datasets used in experimentation and examples of multi-representation views for two collaborative users (a) The *gun turret* dataset (b) The *motorcycle* dataset.

- *Monotonically increasing*, because it is intuitive that individual benefit should increase as the group constraints are lifted, i.e., the coupling is reduced, and vice versa.

These features are desirable even without knowing how useful those objective functions are in real applications.

We implemented a testbed application that enables us to simulate the interactions among different users sharing a 3D model. Each 3D model is composed of many small parts, each having several representations, generated beforehand using a simplification algorithm, e.g., [18]. The collaborative configuration comprised two users—*user1* and *user2*—with *user2* having available three times as much resources as *user1*. We can think of *user1* as collaborating from a mobile device, while *user2* collaborates from a desktop computer. We used a number of 3D objects, downloaded freely from the Internet, and tested the different coupling levels between the users having dissimilar preferences. Figure 5 shows two of the datasets used in the experiment, the *gun turret* (dataset 1) and the *motorcycle* (dataset 2). The coupling preferences were specified as spheres surrounding interesting (parts of) objects, as shown in Figures 3 and 4. We chose their regions of interest so to avoid a significant overlap. Otherwise the coupling can be determined trivially on the shared region of interest, but this fails to account for the general case of dissimilar preferences.

We wanted to compare the results of applying optimization using both SAD and SSD methods described above, for different coupling settings. This coupling is specified as link strength, computed using Eq. (7). For each coupling setting we plot the two terms of the objective functions in Eqs. (5) and (6): the first term represents the local utility and the second term represents the dissimilarity of the representations. The local utility plot shows how the benefit improves for the individuals, while the dissimilarity plot shows how the representations for the two users diverge from one another.

Figure 6 shows the results for dataset 1. Figure 6 (a) shows the local utility and dissimilarity plots for SAD, while Figure 7(b) shows those plots for SSD. Notice the logarithmic scale for the weight ratio ($1/LS$). As this ratio goes to zero, maximum coupling is attained, which results in minimum difference. As the ratio goes to infinity (represented at the point 10000), minimum coupling is reached. Similarly, Figure 7 shows the results for dataset 2. The problem of non-monotonic behavior of the objective function in Figure 7(a) for $1/LS$ in the range [10, 20] is because the problem space is not convex. In such cases, the algorithm selects a
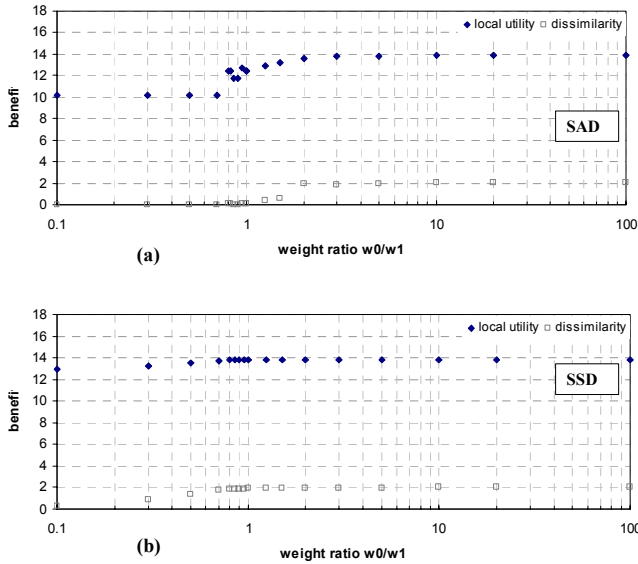
Figure 6. SAD vs SSD performance for Dataset 1.

Figure 7. SAD vs SSD performance for Dataset 2.

representation requiring smaller resources because the resource slack is not sufficient to fit in the next best representation.

The experiments indicate that SSD allows smoother control over the strength of the influence links than SAD, as the weight parameters are modified. This means that for a small change in the weight parameters, users can balance between their coupling settings without abrupt changes in their respective views.

6.1 Resource Efficiency

Coupling the representations for users with disparate resources may prevent the users with greater resources from using those efficiently. Although the data adaptation process for user i is presented with a *nominal* maximum of available resources R_i (as in Eq. (2)), the actual resource usage after adaptation may be lower. This is the penalty of enforcing a tight coupling between devices with scarce resources and more resourceful devices.

We also measured the resource efficiency as the relative error of the actual resources obtained by the optimization framework, and the limit set up by the system:

$$RE_i = \frac{\left| \sum_{j \leq M, k \leq L} r_{i,j,k} x_{i,j,k} - R_i \right|}{R_i}$$

Figure 8 shows the resource efficiency for two users sharing dataset 2, using the same settings as described above. Figure 8(a) shows the resource efficiency for *user1*, i.e., the user with lower capabilities, while Figure 8(b) shows the resource efficiency for *user2*. As expected, the user with the lower resources achieves more efficient resource usage, since his/her resources determine the upper limit when coupling is enforced. Conversely, the users with higher resources get penalized under coupling since some of their resources end up unused. SAD shows non-monotonicity similar to Figure 7(a) due to the non-convex problem space. SSD shows better efficiency for both types of users. This indicates that SSD is a better model for optimizing resources under different coupling requirements than SAD.
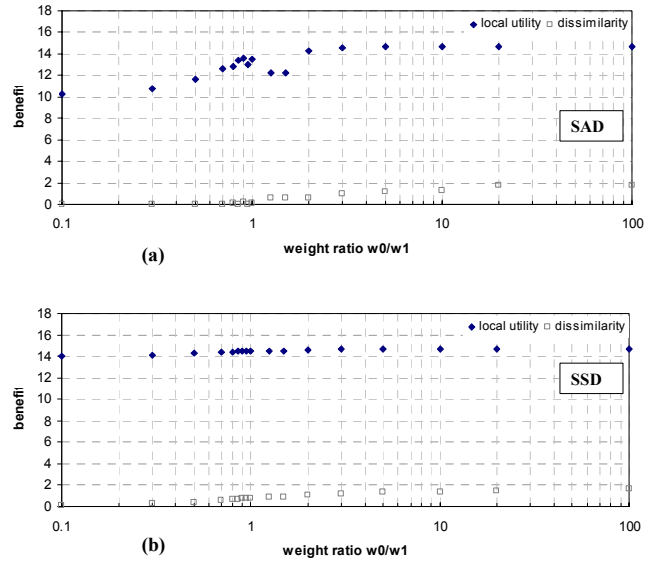
7. CONCLUSIONS

We presented an optimization approach for providing different degrees of coupling in heterogeneous collaborative systems. In such applications, users have different preferences and resource availability, and it is necessary to adapt data to support meaningful collaboration. Our optimization approach guides the data adaptation process to select the best representation for each of the shared objects, so that both individual and group preferences are accounted for. We model group preference as influence links of varying strength. A strong link between two users implies high coupling, while a weak link indicates the lack of coupling. We have shown that finding the best representations can be defined as an optimization problem, where link strength can be specified as a single parameter to represent coupling. We have also presented three different methods for including these coupling parameters, and compared their results in a test-bed collaborative scenario. The results suggest that soft-constraints on coupling yield better satisfaction of the individual and group preferences than hard constraints, and provide a mechanism to smoothly switch between high and low coupling among users working on unequal platforms and having disparate preferences. Evaluation of the impact and usefulness of this framework in real collaborative scenarios is part of the future work.

8. REFERENCES

[1] R. Azuma, Y. Baillot, R. Behringer, S. K. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34-47, 2001.

[2] L. Beca, "Applications of XML and customizable components in building virtual places on the Web," *Proc. IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2000)*, pp. 242-247, 2000.

[3] K. H. Britton, R. Case, A. Citron, R. Floyd, Y. Li, C. Seekamp, B. Topol, and K. Tracey, "Transcoding: Extending e-business to new environments," *IBM Systems J.*, vol. 40, no. 1, pp. 153-178, 2001.

[4] C. D. Cera, T. Kim, J. Han, and W. C. Regli, "Role-based viewing envelopes for information protection in collaborative modeling," *J. of Computer-Aided Design*, Elsevier, vol. 36, pp. 873-886, 2004.
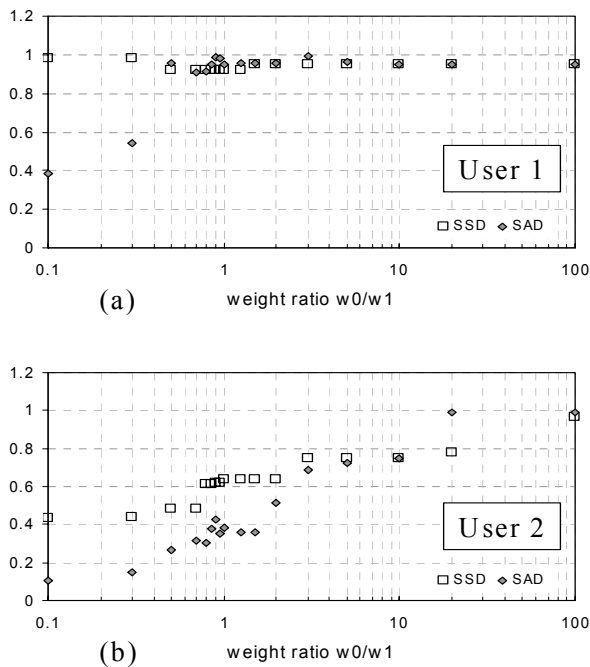
Figure 8. Resource efficiency for (a) user 1 (b) user 2.

[5]   C. D. Correa and I. Marsic, "Software framework for managing heterogeneity in mobile collaborative systems," *Computer Supported Cooperative Work J.*, vol. 13, no. 5-6, pp. 603-638, 2004

[6]   M. Deering, "Geometry compression," *Proc. ACM SIGGRAPH 95*, Los Angeles, CA, pp. 13-20, 1995.

[7]   E. De Lara, R. Kumar, D. S. Wallach, and W. Zwaenepoel, "Collaboration and multimedia authoring on mobile devices," *Proc. 1st Int'l Conf. Mobile Systems, Applications, and Services (MobiSys'03)*, San Francisco, CA, pp. 287-301, 2003.

[8]   P. Dewan and R. Choudhary, "Flexible user interface coupling in a collaborative system," *Proc. ACM Conf. Human Factors in Computing Systems (CHI'91)*, New Orleans, LA, pp. 41-48, 1991.

[9]   S. K. Feiner, "Augmented reality: A new way of seeing," *Scientific American*, vol. 286, no. 4, pp. 34-41, April 2002 .

[10] S. K. Feiner, H. Fuchs, T. Kanade, G. Klinker, P. Milgram, and H. Tamura, "Mixed reality: Where real and virtual worlds meet," *Proc. ACM SIGGRAPH 99 Conference Abstracts and Applications*, Los Angeles, CA, pp.156-158, August 1999.

[11] J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics: Principles and Practice*. Addison Wesley, 1993.

[12] A. Fox, S. D. Gribble, Y. Chawathe, and E. A. Brewer, "Adapting to network and client variation using active proxies: Lessons and perspectives," *IEEE Personal Communications*, vol. 5, no. 4, pp. 10-19, 1998.

[13] A. Fox, S. D. Gribble, E. A. Brewer, and E. Amir, "Adapting to network and client variability via on-demand dynamic distillation", *Proc. 7th Conf. Architectural Support for Programming Languages and Operating Systems*, Cambridge, MA, pp.160-170, 1996.

[14] T. Funkhouser and C. H. Sequin, "Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments," *Proc. ACM SIGGRAPH Computer Graphics Annual Conf.*, Los Angeles, CA, pp. 99-108, 1993.

[15] C. Gutwin and S. Greenberg, "Design for individuals, design for groups: Tradeoffs between power and workspace awareness," *Proc. ACM Conf. Computer Supported Cooperative Work (CSCW'98)*, Seattle, pp. 207-296, 1998.

[16] R. W. Hall, A. Mathur, F. Jahanian, A. Prakash, and C. Rassmussen, "Corona: A communication service for scalable, reliable group collaboration systems," *Proc. ACM Conf. Computer Supported Cooperative Work (CSCW'96)*, Boston, MA, 1996.

[17] T. H. Höllerer, S. Feiner, D. Hallaway, B. Bell, M., Lanzagorta, D. Brown, S. J. Julier, Y. Baillot, and L. J. Rosenblum, "User interface management techniques for collaborative mobile augmented reality," *Computers & Graphics*, vol. 25, no. 5, pp. 799-810, 2001.

[18] H. Hoppe, "Progressive meshes," *Proc. ACM SIGGRAPH Computer Graphics Annual Conf.*, New Orleans, LA, pp. 99-108, 1996.

[19] J. P. Ignizio and T. M. Cavalier, *Linear Programming*. Englewood Cliffs, NJ: Prentice Hall, 1994.

[20] S. Julier, M. A. Livingston, J. E. Swan II, Y. Baillot, and D. Brown, "Adaptive user interfaces in augmented reality," *Proc. Workshop on Software Technology for Augmented Reality Systems*, Tokyo, 2003.

[21] M. Klein, H. Sayama, P. Faratin, and Y. Bar-Yam, "What complex systems research can teach us about collaborative design," *Proceedings of the 6th Int'l Conf. Computer Supported Cooperative Work in Design (CSCWD-2001)*, 5-12, 2001.

[22] M. J. Knister and A. Prakash, "DistEdit: A distributed toolkit for supporting multiple group editors," *Proc. ACM Conf. Computer Supported Cooperative Work (CSCW'90)*, Los Angeles, CA, pp. 343-355, 1990.

[23] D. Li and L. Rui, "Transparent sharing and interoperation of heterogeneous single-user applications," *Proc. ACM Conf. Computer Supported Cooperative Work (CSCW'02)*, New Orleans, LA, pp. 246-255, 2002.

[24] W. Y. Lum and F. C. M. Lau, "On balancing between transcoding overhead and spatial consumption in content adaptation," *Proc. 8th Int'l Conf. Mobile Computing and Networking (MobiCom '02)*, Atlanta GA, pp. 239-250, 2002.

[25] I. Mani, *Automatic Summarization*. John Benjamins Publ. Co., Amsterdam/Philadelphia, 2001.

[26] D. Marcu, *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press, Cambridge, MA, 2000.

[27] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc, 1990.

[28] Moving Picture Experts Group. http://www.mpeg.org

[29] B. D. Noble, M. Price and M. Satyanarayanan, "A programming interface for application-aware adaptation in mobile computing," *Proc. 2nd Symp. Mobile and Location-Independent Computing*, pp. 57-66, April 1995.

[30] A. Prakash and H. Shim, "DistView: Support for building efficient collaborative applications using replicated objects," *Proc. ACM Conf. Computer Supported Cooperative Work (CSCW'94)*, pp. 153-164, 1994.

[31] T. Phan, G. Zorpas, and R. Bagrodia, "Middleware support for reconciling client updates and data transcoding," *Proc. 2nd Int'l Conf. Mobile Systems, Applications, and Services (MobiSys'04)*, Boston, MA, pp. 139-152, 2004.

[32] D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman, "Navigating hierarchically clustered networks through fisheye and full-zoom methods," *ACM Transactions on Computer-Human Interaction*, vol. 3, no. 2, pp. 162-188, June 1996.

[33] S. Shirmohammadi, A. El Saddik, N. D. Georganas, and R. Steinmetz, "JASMINE: A Java tool for multimedia collaboration on the Internet," *J. Multimedia Tools and Applications*, vol. 19, no. 1, pp. 5-28, 2003.

[34] M. Stefik, D. Bobrow, S. Lanning and D. Tator, "WYSIWIS revised: Early experiences with multiuser interfaces," *ACM Trans. Information Systems*, vol. 5, no. 2, pp. 147-167, 1987.

[35] Web3D Consortium. VRML97 and Related Specifications. Online at: http://www.web3d.org/x3d/specifications/vrml/

[36] E. A. Wernert and A. J. Hanson, "A framework for assisted exploration with collaboration," *Proc. IEEE Visualization '99*. pp 241-248, 1999.

[37] M. Yokoo and K. Hirayama. "Algorithms for distributed constraint satisfaction: A review". *Autonomous Agents and Multiagent Systems*, vol. 3, no. 2, pp. 189-211, 2000.