

## 6 Conclusão

Neste trabalho, foi apresentado um sistema colaborativo capaz de controlar as versões das edições de um vídeo no formato MPEG-2, sem que os editores estejam no mesmo local, ao mesmo tempo. Este capítulo faz o fechamento da dissertação, apresentando as principais contribuições e propondo trabalhos futuros.

### 6.1. Contribuições da Dissertação

As principais contribuições desta dissertação foram: definir uma estrutura de dados que permita o controle de versão das edições cooperativas de vídeo no formato MPEG-2; especificar um sistema colaborativo de edição de vídeo entre diferentes editores; definir um mecanismo de remontagem do vídeo no domínio comprimido; e implementar o primeiro protótipo do sistema de controle de versão para edição cooperativa de vídeo.

A definição de uma estrutura de dados para prover o controle de versão das edições cooperativas de vídeo permitiu o gerenciamento e a recuperação das edições do vídeo versionado, no decorrer do tempo. O conteúdo audiovisual foi organizado numa estrutura de dados chamada de *árvore de versionamento*, cujas sub-árvores de cada nó estão ordenadas, e as folhas descrevem um conjunto ordenado de trechos do vídeo. Essa definição também permite o compartilhamento de versões entre as árvores de versionamento e a extração de informações de autoria de cada versão.

A especificação de um sistema de edição colaborativa de vídeo permite a divisão de tarefas e o trabalho concorrente no processo de produção entre os editores. A partir dessa especificação, foi desenvolvido um sistema distribuído capaz de controlar os acessos simultâneos da árvore de versionamento e realizar a fusão das contribuições das versões. O sistema permite que usuários acessem versões de um vídeo ou trecho dele, mesmo estando dispersos geograficamente.

Apesar de não ter sido um objetivo deste trabalho, um mecanismo de segmentação do vídeo MPEG-2 foi implementado no domínio comprimido para permitir aos editores a manipulação dos segmentos, de forma independente. O tratamento das bordas foi utilizado quando ocorrem dependências de quadros nas bordas dos segmentos adjacentes entre si. A duplicação de nós dependentes das bordas entre os segmentos consecutivos é a técnica utilizada para que os segmentos possam ser visualizados corretamente nos *players*. Um algoritmo de detecção automática de tomadas foi apresentado, embora a implementação atual do sistema não contemple tal funcionalidade.

Outro mecanismo apresentado foi o da remontagem dos segmentos do vídeo MPEG-2, também no domínio comprimido. Este mecanismo concatena os vídeos das folhas da árvore de versionamento, na composição da versão final do vídeo comprimido.

A implementação do sistema foi desenvolvida sobre a plataforma Windows e Linux, utilizando a linguagem de programação Java, bibliotecas padronizadas e CORBA. O protótipo implementado, denominado *VideoCVS*, é um sistema distribuído baseado na arquitetura de comunicação cliente e servidor. Foram desenvolvidos os componentes do cliente e servidor para armazenar as versões e os arquivos audiovisuais, e o gerenciador das requisições dos clientes. Foi feita a integração entre o *VideoCVS* e ferramentas de edição de vídeo.

## **6.2. Trabalhos Futuros**

Dentre as principais características a serem incorporadas ao sistema *VideoCVS*, em trabalhos futuros, estão: a inclusão do mecanismo de detecção de tomadas de cenas automático; o suporte a outros tipos de formatos de mídias, como MPEG-4; outros meios de tratamento das bordas dos segmentos e controle de buffer quando incorporados os mecanismos de segmentação e remontagem do arquivo audiovisual; o estudo de mecanismos de fusão entre arquivos audiovisuais no formato MPEG; a inclusão de anotações em vídeo; o controle de acesso baseado em papéis e a notificação de modificações das árvores de versionamento entre os editores, em tempo real; o armazenamento das operações realizadas sobre

os vídeos da árvore; e o aperfeiçoamento do protótipo quanto às funcionalidades e interface gráfica.

A utilização do mecanismo de detecção automática de tomadas de cenas no *VideoCVS* poderia trazer contribuições e refinamentos na estruturação da árvore de versionamento. O mecanismo seria essencial para indexação, recuperação e navegação (*browsing*) dos acervos de vídeo.

O padrão de vídeo MPEG-4 é hoje uma das apostas para as próximas gerações de aplicações multimídia (TV Digital, Ambientes Hipermídias) da indústria. Diferente do padrão MPEG-2, que foca principalmente nos aspectos relacionados à codificação e decodificação linear de vídeo e áudio, o MPEG-4 define uma codificação baseada em objetos e possui algumas vantagens como: o reuso do material audiovisual, dada a abordagem orientada à objeto do padrão; a possibilidade de codificação dos objetos usando diferentes resoluções espaciais e temporais; a possibilidade de uso de técnicas de compactação e compressão que podem ser aplicadas individualmente, de acordo com as características de cada mídia; e a utilização de objetos sintéticos, que variam desde objetos simples de duas dimensões, como linhas e pontos, até objetos complexos de três dimensões, como animações faciais e corporais, em conjunto com os demais objetos de mídia naturais, como câmeras, microfones etc.

Embora o *VideoCVS* realize a segmentação e a remontagem de arquivos MPEG-2 vídeo no domínio comprimido, o que garante a qualidade do vídeo original, em termos de confiabilidade o mecanismo pode não ser satisfatório. A duplicação de quadros nas bordas dos segmentos dependentes entre si, sem o controle de bits de cada quadro pode gerar overflow para segmentos de tamanhos grandes e underflow para segmentos de tamanhos pequenos. Para otimizar esses mecanismos, é preciso manipular corretamente a quantidade de bits de cada quadro, a taxa de bits do fluxo codificado e o tamanho do buffer, depois das edições de cada segmento e na remontagem deles. Essa reformulação de bits, taxas e tamanho do buffer são operações que necessitam de estudos mais aprofundados, visto que devem ser realizadas em tempo real, no fluxo MPEG de sistemas e no domínio comprimido.

O mecanismo de fusão entre dois arquivos de vídeo poderia produzir um único arquivo de vídeo resultante. No *VideoCVS*, esse mecanismo é feito entre duas versões da árvore de versionamento, que produz uma árvore resultante, mas

nunca dois arquivos MPEG-2 de vídeo são juntados em um único arquivo de vídeo. Quando há nós considerados equivalentes derivados mas com timestamp maior que o nó de origem, uma intervenção manual é solicitada pelo sistema. Caso houvesse um mecanismo de fusão entre os conteúdos dos nós, esse mecanismo poderia substituir a intervenção manual. Como o formato MPEG-2 tem uma estrutura, seria interessante estudar a possibilidade de realizar esse mecanismo, comparando-se as diferenças entre os arquivos. A fusão entre arquivos MPEG-2 poderia permitir que usuários modificassem uma mesma folha de uma árvore de versionamento, em paralelo.

Uma busca refinada dos conteúdos dos nós poderia estar presente no VideoCVS. Como o VideoCVS não implementou o conceito de anotação de vídeo, o sistema é incapaz de permitir que usuários possam facilmente encontrar e também definir determinados tipos, categorias, tamanhos, resoluções de vídeos, por exemplo. A inclusão de anotações em vídeo poderia ser realizada, visto que a visualização dos segmentos anotados e também a consulta e navegação em seu conteúdo indexadas pela lista de anotações poderiam trazer benefícios aos editores. As anotações poderiam ser definidas como comentários inseridos em regiões que são destacadas em um ou mais quadros do vídeo. E essa inclusão poderia ser feita a partir da extração automática da semântica de um vídeo (ou de parte dele) ou mesmo manualmente.

O primeiro protótipo do VideoCVS não gerencia quais usuários estão modificando uma determinada árvore de versionamento nem possui um controle de acesso baseado em papéis. Uma notificação de modificações entre usuários poderia permitir que houvesse uma maior colaboração na edição de uma árvore de versionamento. Em tempo real, o sistema não possui uma rotina de verificação de quais usuários estão modificando um determinado nó da árvore. Como é possível que alguns nós possam ser bloqueados, e que os bloqueadores possam esquecer de desbloqueá-los, o sistema poderia estipular tempos de expiração de bloqueios para determinados grupos de usuários (papéis), e alguns grupos poderiam ter prioridade, de bloqueios e desbloqueios, sobre outros.

O VideoCVS não tem domínio sobre quais e onde foram as operações e modificações feitas em uma árvore de versionamento. A integração entre uma ferramenta de edição e o VideoCVS existe, mas ainda longe de ser considerada uma integração fortemente acoplada. Possivelmente, para prover essa integração

satisfatória, seja necessário implementar dentro do VideoCVS uma ferramenta de edição ou então incorporar às ferramentas, o VideoCVS.

Novas funcionalidades poderiam ser introduzidas no protótipo, afim de melhorar a interface gráfica para facilitar o uso do sistema. Funcionalidades como: a possibilidade de cortar uma sub-árvore e colar, esta mesma sub-árvore, em outra posição de uma árvore de versionamento; chat entre os usuários do sistema; a possibilidade de visualizar a hierarquia de derivação das árvores de versionamentos e dos nós, através de grafos; entre outros.