

## Capítulo

# 7

## Projetos Atuais e Visão de Futuro do Laboratório TeleMídia/PUC-Rio em Videocolaboração

Sérgio Colcher<sup>1</sup>, Alan L. V. Guedes<sup>1</sup>, Roberto Gerson de A. Azevedo<sup>1</sup>,  
Guilherme F. Lima<sup>1</sup>, Rodrigo C. M. Santos<sup>1</sup>, Antonio J. G. Busson<sup>1</sup>

<sup>1</sup>Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

{colcher, aguedes, razevedo, glima, rsantos, abusson}@inf.puc-rio.br

### *Abstract*

*Since the early 1990's, the TeleMídia laboratory of the Department of Informatics of PUC-Rio researches and trains people in the areas of multimedia and hypermedia, which include interactive videos and video collaboration as important subareas. This chapter presents the main research lines and projects of the laboratory and discusses the future vision of the authors about interactive videos and video collaboration.*

### *Resumo*

*Desde a década de 1990 o laboratório TeleMídia do Departamento de Informática da PUC-Rio realiza pesquisas e forma recursos humanos em sistemas multimídia e hipermídia, dos quais vídeos interativos e videocolaboração são importantes subáreas. Este capítulo apresenta as principais linhas e projetos de pesquisa do laboratório e a visão de futuro dos autores sobre vídeos interativos e videocolaboração.*

### **7.1. Introdução**

Desde 1991 o laboratório TeleMídia, um dos laboratórios temáticos do Departamento de Informática da PUC-Rio, desenvolve pesquisas e forma recursos humanos em sistemas multimídia e hipermídia. Durante esses anos, vários têm sido os temas de pesquisa abordados pelo laboratório, incluindo diversas aplicações de sistemas multimídia, tais como sistemas de teleconferência, e-mail multimídia, TV e rádio digitais, IPTV, streaming de mídia sob demanda, e, mais recentemente, experiências imersivas e internet das coisas. Vários desses trabalhos estão relacionados aos temas de vídeos interativos e videocolaboração, sendo que muitas das tecnologias e dos padrões desenvolvidos no laboratório podem servir de base para a implementação de diferentes cenários de aplicações avançadas nestas áreas.

Além do seu papel no desenvolvimento científico e na formação de pessoas, o laboratório TeleMídia também tem participado ativamente na transferência de tecnologia para a indústria e contribuído com órgãos de padronização nacionais e internacionais, tais como a Associação Brasileira de Normas Técnicas (ABNT) e a União Internacional das Telecomunicações (UIT). No que se refere à padronização, o laboratório tem como destaques a criação do *middleware* Ginga e da linguagem NCL que suportam o desenvolvimento de aplicações multimídia interativas para ambientes de TV digital, IPTV e sistemas integrados *broadcast/broadband* (IBB). Tanto o *middleware* Ginga como a linguagem NCL são hoje padrões ABNT para sistemas de TV digital terrestre [ABNT 2016] e recomendações UIT-T para sistemas de IPTV e sistemas IBB [ITU-T 2014].

Este capítulo apresenta as principais linhas de pesquisa do TeleMídia e detalha os projetos em andamento relacionados ao tema de videocolaboração (Seção 1.2). Além disso, o capítulo apresenta a visão de futuro dos membros do laboratório (Seção 1.3) sobre como avanços tecnológicos recentes podem viabilizar novos cenários de videocolaboração, e sobre como cenários hoje inviáveis podem motivar o desenvolvimento de tecnologias ainda inexistentes. Ao final do capítulo, são apresentadas as conclusões (Seção 1.4).

## **7.2. Linhas de pesquisa e projetos atuais**

O laboratório TeleMídia possui três linhas de pesquisa relacionadas à videocolaboração: (1) *Representação e autoria de cenas multimídia interativas*; (2) *Arquitetura e implementação de middlewares para sistemas multimídia*; e (3) *Experiências imersivas*. Esta seção apresenta os objetivos, resultados e possíveis aplicações dos trabalhos desenvolvidos em cada uma dessas linhas no contexto de vídeos interativos e de videocolaboração.

### **7.2.1. Representação e autoria de cenas multimídia**

O objetivo desta linha de pesquisa é desenvolver modelos, linguagens e ferramentas para a autoria de aplicações multimídia interativas. Uma aplicação multimídia interativa é composta pela apresentação sincronizada de múltiplos objetos de mídia (textos, áudios, vídeos, animações e, mais recentemente, mídias não-tradicionais, como mídias olfativas, táctis, etc.) e responde à interação do usuário (ou usuários, no caso de aplicações multiusuário).

Modelos e linguagens projetados especificamente para a construção de aplicações multimídia interativas (ou *modelos e linguagens multimídia*) são uma base adequada para a especificação de diversos tipos de aplicações de vídeo interativo e videocolaboração. A principal vantagem de se usar esses modelos e linguagens em vez de linguagens de programação de propósito geral, é que, em geral, eles oferecem abstrações de alto nível mais próximas do domínio de aplicações multimídia interativas.

Por exemplo, em uma linguagem multimídia, especificar o sincronismo entre o vídeo principal da aplicação e objetos de mídia secundários é normalmente uma tarefa simples, já que a linguagem oferece construções deliberadamente projetadas para esse fim. (Em um contexto de videocolaboração, podemos imaginar que o vídeo principal é aquele compartilhado entre os usuários colaboradores, e os objetos de mídia secundários representam os controles necessários para realizar uma determinada tarefa.)

Por outro lado, implementar cenários similares em uma linguagem de programação de propósito geral exige que o programador escreva essa mesma especificação em um nível de abstração mais baixo, utilizando primitivas mais próximas do modelo de execução da máquina, e distantes do domínio da aplicação. Assim, a tendência é que o programador acabe “reinventando a roda”. Isto é, em vez de trabalhar diretamente na especificação da solução do problema (no caso, na especificação do sincronismo entre o vídeo e os objetos secundários) usando um vocabulário no nível de abstração adequado, ele acaba implementando esse vocabulário.

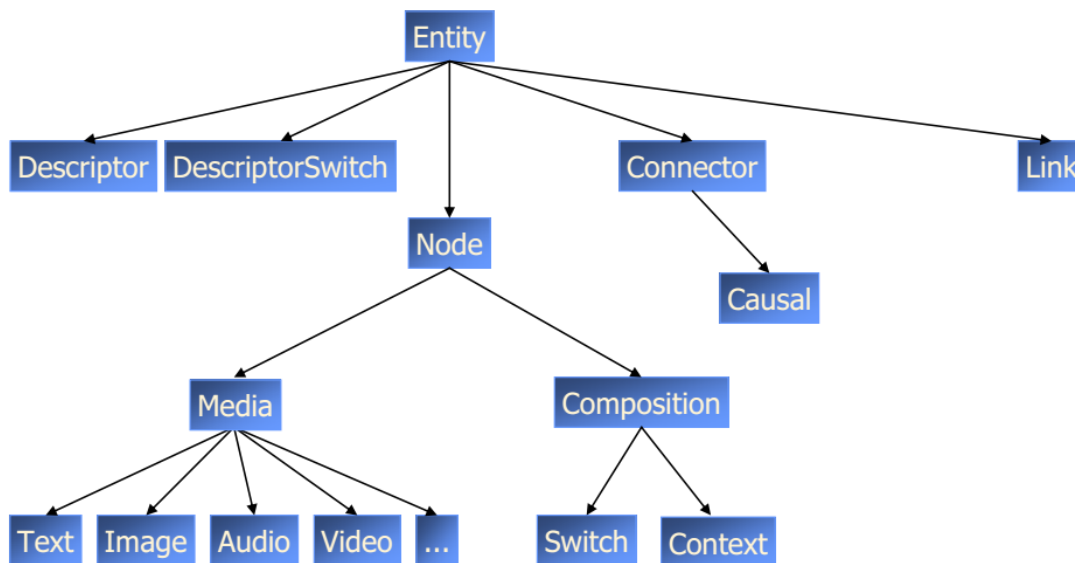
Há na literatura inúmeros exemplos de aplicações baseadas em vídeo que tiram proveito de linguagens e modelos multimídia. Algumas são aplicações locais para construção e reprodução de artefatos interativos em um mesmo dispositivo [Meixner 2017], enquanto outras são aplicações distribuídas onde diversos usuários colaboram simultaneamente para a realização de uma determinada tarefa [Jansen et al. 2015].

No caso de aplicações de vídeo interativo e de videocolaboração, os requisitos de distribuição de conteúdo e de controle tornam particularmente atrativos o uso de modelos e linguagens com suporte a múltiplos dispositivos [Van Deventer et al. 2016] e múltiplos usuários [Guedes, Á. L. et al. 2016]. Um dos modelos que possui tais características é o NCM, base conceitual da linguagem NCL. Ambos, NCM e NCL, foram desenvolvidos no laboratório TeleMídia.

O restante desta seção apresenta o NCM e a NCL e discute a aplicabilidade dessas tecnologias no contexto de videocolaboração. Em seguida, são apresentadas as principais ferramentas de apoio à criação de aplicações NCL desenvolvidas no laboratório: NCL Eclipse, ISB Designer e NCL Composer.

### ***NCM e NCL***

O modelo NCM (*Nested Context Model* ou Modelo de Contextos Aninhados) [Casanova et al. 1991], cuja primeira versão data de 1991, é um modelo conceitual para a especificação e apresentação de documentos hipermídia. O NCM combina as ideias usuais de nós e elos hipermídia com o conceito de contextos aninhados, que permite a definição de escopo e uma melhor organização das informações representadas. A Figura 1 apresenta as principais entidades do NCM 3.0 (versão atual do modelo).



**Figura 1. Entidades básicas do NCM 3.0** [Luiz Fernando Gomes and Barbosa 2009].

NCL (*Nested Context Language* ou Linguagem de Contextos Aninhados) [Soares et al. 2006] é uma linguagem declarativa para a especificação de aplicações multimídia interativas baseada no NCM. A primeira versão da linguagem, NCL 1.0, foi desenvolvida em 2000 [Antonacci et al. 2000], e a versão mais recente, NCL 3.0, foi lançada em 2006 [ABNT 2016]. NCL 3.0 é hoje a linguagem declarativa padrão para a especificação de aplicações interativas no sistema brasileiro de TV digital [ABNT 2016] e recomendação UIT-T para sistemas IPTV [ITU-T 2014].

Um programa NCL 3.0 é um código XML (*eXtensible Markup Language*) que descreve uma apresentação multimídia interativa. Essa descrição consiste, basicamente, da especificação de um conjunto de objetos mídia (textos, imagens, áudios, vídeos, etc.) e de um conjunto de elos de sincronismo, que determinam o comportamento dos objetos de mídia durante a apresentação. Objetos de mídia e elos agrupados em contextos, os quais podem ser aninhados, possibilitando uma melhor organização do código. Em NCL 3.0, os elos são relacionamentos causais, isto é, pares da forma condição-ação que relacionam eventos.

Além de primitivas para descrição do conteúdo (mídias) e do comportamento de uma aplicação multimídia, NCL 3.0 possui funcionalidades adicionais particularmente úteis para o desenvolvimento de aplicações de videocolaboração. A primeira delas é o suporte à aplicações cuja apresentação pode ser distribuída em múltiplos dispositivos em uma rede local. Em NCL 3.0, esse tipo de aplicação é especificada de forma declarativa—o autor da aplicação precisa apenas especificar as classes de dispositivos suportadas e o em qual classe de dispositivo cada objeto será apresentado [Batista 2013; Soares et al. 2009]. O suporte da linguagem à múltiplos dispositivos é discutido em detalhe na Seção 1.2.3.

A segunda funcionalidade de NCL 3.0 interessante para o desenvolvimento de aplicações de videocolaboração é o suporte da linguagem à comandos de edição. Comandos de edição são instruções que modificam dinamicamente o código ou o estado

de uma aplicação NCL. Há comandos para inserir ou remover objetos de mídia, inserir ou remover elos, iniciar ou parar partes específicas da aplicação, etc. Tais comandos podem ser gerados tanto pela emissora (no caso de aplicações de TV digital terrestre ou IPTV) quanto programaticamente usando *scripts* Lua [Sant'Anna et al. 2008], a linguagem de *script* de NCL. Adicionalmente, também é possível usar Lua para enviar e receber dados diretamente via TCP/IP e HTTP [Sant'Anna et al. 2008]. Tais funcionalidades, possibilitam, por exemplo, o desenvolvimento de aplicações de TV Social [Cesar and Geerts 2011; Fehine 2017], onde usuários podem colaborar enquanto assistem um determinado programa de TV, e.g. comentando

Também podem ser úteis para o desenvolvimento de aplicações de videocolaboração extensões recentes (ainda não padronizadas) que adicionam à NCL 3.0 suporte à especificação declarativa de mídias estereoscópicas [Azevedo 2015] e interação multimodal [Guedes, A. L. et al. 2016; Guedes, Á. L. et al. 2016]. Ambas essas extensões são resultado de teses desenvolvidas recentemente no laboratório TeleMídia e são detalhadas na Seção 1.2.3.

Além de desenvolver pesquisas relacionadas à extensão da NCL para novos cenários de aplicação, o laboratório TeleMídia mantém um projeto dedicado à evolução do padrão ABNT e da recomendação UIT-T da NCL. O objetivo principal desse projeto é desenvolver o perfil completo e o perfil restrito da versão 4.0 de NCL.

*O perfil completo* da NCL 4.0 deverá incluir:

1. um melhor suporte à aplicações cientes de contexto;
2. um melhor suporte à aplicações distribuídas em múltiplos dispositivos, incluindo um melhor suporte à serviços de TV social;
3. suporte declarativo ao desenvolvimento de aplicações com conteúdo dinâmico;
4. suporte à inclusão de objetos e eventos 3D;
5. suporte à relacionamentos de restrição, em adição aos de causalidade;
6. suporte a templates (integrando as funcionalidades de TAL [Soares Neto et al. 2012]).

Já o *perfil restrito* da NCL 4.0, chamado de Raw Profile, deverá incluir:

1. a especificação do perfil de NCL 4.0 excluindo todos os elementos redundantes do perfil completo da NCL 4.0 que possam afetar o desempenho e robustez do *player* NCL;
2. o desenvolvimento de um conversor do perfil completo para o perfil Raw;
3. o desenvolvimento de conversores de outras linguagens declarativas para a criação de aplicações multimídia (e.g. SMIL [Bulterman and Rutledge 2016] e ) para o perfil NCL 4.0 Raw;
4. a implementação de referência do Ginga-NCL seguindo o perfil Raw;
5. o desenvolvimento de conversores do perfil NCL 4.0 Raw para exibição em *players* de outras linguagens declarativas.

A maioria dos itens acima, em especial os do perfil completo, estão relacionados ao contexto de videocolaboração. Por exemplo, melhor suporte à aplicações cientes de

contexto implica em aplicações que se adaptam melhor às peculiaridades de cada usuário, o que é um requisito importante para aplicações multiusuário. Um argumento similar se aplica ao melhor suporte à aplicações distribuídas—um sistema de videocolaboração remota é, por definição, um sistema distribuído.

### ***NCL Eclipse, ISB Designer e NCL Composer***

As principais ferramentas de autoria multimídia desenvolvidas pelo laboratório TeleMídia são o NCL Eclipse, o ISB Designer e o NCL Composer (vide Figura 2).

O *NCL Eclipse* [Azevedo et al. 2011] é um *plugin* para a IDE Eclipse<sup>1</sup> que provê facilidades para ajudar programadores a desenvolver aplicações NCL. Essas facilidades incluem coloração de código, sugestão de código (*autocomplete*), navegação no código por meio de hiper-elos, visualização programática, verificação sintática e integração com diferentes *players* NCL. Atualmente, o NCL Eclipse é a ferramenta mais utilizada na América Latina para o desenvolvimento de aplicações interativas para o *middleware* Ginga.

O *ISB Designer* [Araújo and Soares 2014] é uma ferramenta de apoio ao projeto de narrativas interativas baseada no conceito de *storyboards* interativos—uma extensão dos *storyboards* tradicionais utilizados por *designers* e profissionais de TV para a descrição de narrativas. A vantagem dos *storyboards* interativos é que eles são fáceis de editar, e permitem que o autor construa narrativas não-lineares de forma gradual (contendo artefatos ainda incompletos) e independente de uma linguagem-alvo específica. O *ISB Designer* também permite a edição e visualização de cadeias temporais interligadas por eventos não-determinísticos, cada uma delas compondo um *storyboard* com pontos de eventos não-previsíveis (por exemplo, a interação do usuário). A ferramenta permite exportar o projeto para NCL e HTML, e pode ser utilizada de forma *standalone* ou como um *plugin* do NCL Composer [Araújo and Soares 2014].

O *NCL Composer* [Azevedo et al. 2012] é um ambiente de autoria para aplicações NCL que facilita o uso dos conceitos definidos pelo NCM e pela NCL por não-programadores. Além da visão textual, que traz funcionalidades similares às do NCL Eclipse, o NCL Composer também traz diferentes visões gráficas, tais como a visão estrutural e de leiaute, que permitem que usuários não-programadores utilizem abstrações visuais para o desenvolvimento de aplicações multimídia. Cada uma dessas visões é implementada como um *plugin* da ferramenta.

Os seguintes *plugins* compõem a distribuição oficial do NCL Composer:

- *Visão textual*: provê as mesmas funcionalidades hoje oferecidas pela ferramenta NCL Eclipse para a edição do código do documento.
- *Visão estrutural*: oferece uma representação gráfica da estrutura do documento.
- *Visão de leiaute*: permite a edição gráfica das propriedades de posicionamento e dimensionamento de objetos de mídia NCL.
- *Visão de propriedades*: permite a edição das propriedades de objetos de mídia.
- *Plugin de validação*: verifica a sintaxe do documento.

---

<sup>1</sup> Disponível em <http://eclipse.org>.

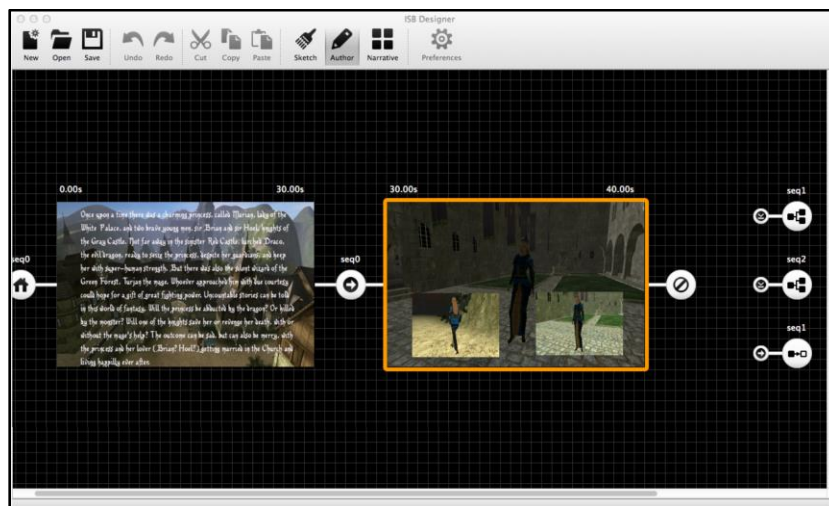
- *Visão de execução:* permite apresentar um documento NCL dentro da própria ferramenta.

```

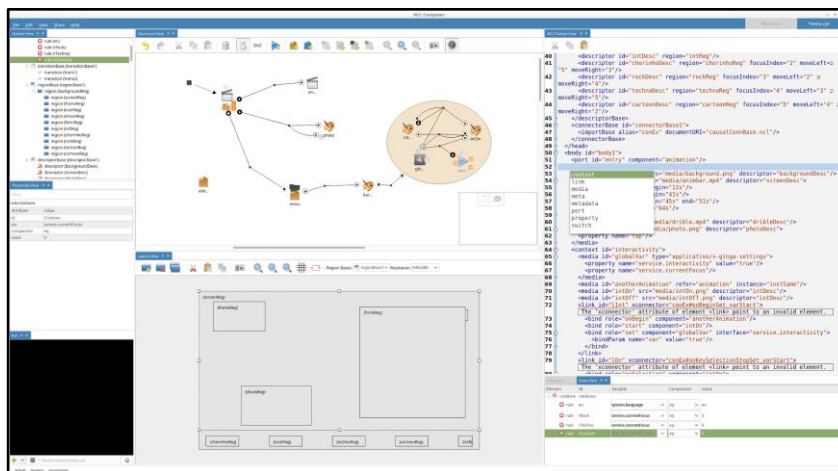
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <!DOCTYPE ncl SYSTEM "http://www.ncl.org.br/NCL1.0/EXTVProfile">
3
4 <!-- CABEÇALHO DO PROGRAMA -->
5
6 <!--
7
8 <!--
9
10 <!--
11
12 <!--
13
14 <!--
15 <!--
16 <!--
17 <!--
18 <!--
19 <!--
20 <!--
21 <!--
22 <!--
23
24
25
26
27
28
29
30
31
32
33
34

```

(a)



(b)



(c)

**Figura 2. Ferramentas de autoria multimídia desenvolvidas no laboratório TeleMídia: (a) NCL Eclipse, (b) ISB Designer, e (c) NCL Composer.**

A arquitetura do NCL Composer é baseada em um núcleo mínimo, que gerencia as mensagens trocadas pelos diferentes *plugins*, e que pode ser estendido por *plugins* de terceiros. Os *plugins* de terceiros podem ser novas visões (gráficas ou textuais) para auxílio à autoria, ou novos geradores de dados em uma dada sintaxe de transferência para os vários sistemas de TV digital, ou em um dado formato de armazenamento. Atualmente, existem diversos *plugins* desenvolvidos por terceiros para facilitar a criação de aplicações interativas de domínio específico [Bernal and Mejía 2015; Ferreira and Souza Filho 2014; Mejía et al. 2014; Santos et al. 2013].

O núcleo do NCL Composer tem se mostrado estável, escalável e flexível. Além da manutenção do núcleo e dos *plugins* oficiais mencionados acima, o laboratório TeleMídia mantém as seguintes frentes de evolução do NCL Composer:

1. Implementação de um *plugin* de visão temporal, isto é, uma visão que represente no tempo os eventos associados à apresentação da aplicação.
2. Implementação de *plugins* para a codificação e transmissão da aplicação em sistemas de TV digital, rádio digital e IPTV.
3. Suporte dinâmico à diferentes perfis de usuários (de especialistas a não-programadores) e ambientes (de *high-end* a *low-end*).
4. Suporte à edição interativa da aplicação, isto é, permitir que o autor modifique-a em tempo de execução, enquanto testa o que já foi implementado.
5. Implementação de um *plugin* para autoria WYSIWYG (*What You See Is What You Get*) de aplicações para múltiplos dispositivos.
6. Suporte à distribuição de *plugins* em múltiplos processos e múltiplas máquinas, com foco em suportar a autoria colaborativa de documentos multimídia.

### **7.2.2. Arquitetura e implementação de sistemas multimídia**

O objetivo desta linha de pesquisa é desenvolver exibidores (*players*) e arquiteturas de referência que permitam executar os modelos e linguagens multimídia apresentados na Seção 1.2.1 de forma correta e eficiente. Também está relacionado a esta linha de pesquisa o desenvolvimento de algoritmos e mecanismos de sinalização e de transporte que permitam obter maior qualidade de experiência (QoE) tanto em aplicações locais quanto distribuídas.

#### ***Middleware Ginga***

As abstrações oferecidas por modelos e linguagens multimídia são úteis para a modelagem teórica de vários tipos de aplicações em videocolaboração. Entretanto, é o desenvolvimento de *players* (ou exibidores) que torna esses modelos e linguagens úteis na prática. Um exibidor de um modelo ou linguagem multimídia é um programa que interpreta as instruções de alto nível do documento de entrada e as transforma em uma apresentação multimídia equivalente.

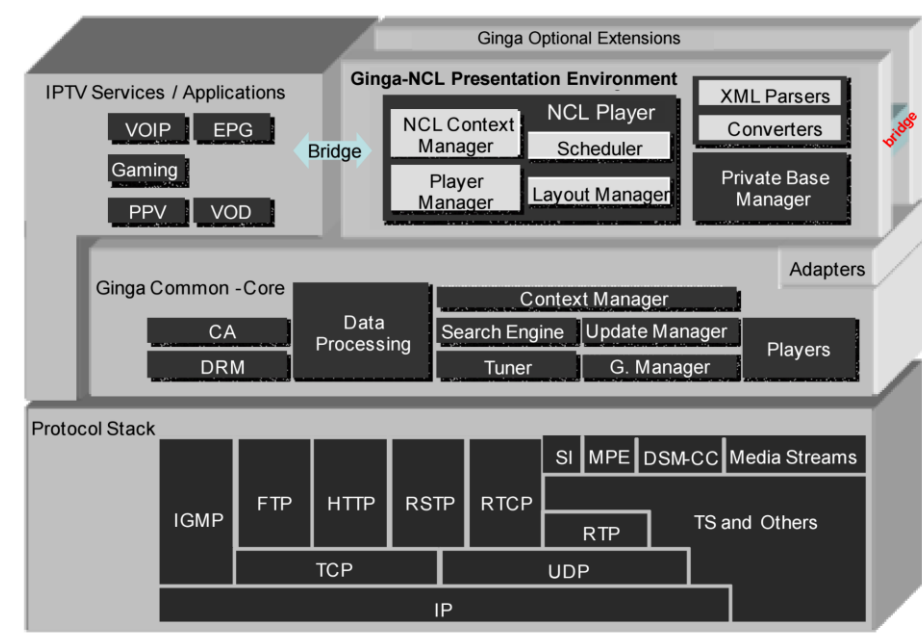
Apesar de essencial, o exibidor é apenas um dos componentes de uma solução multimídia integrada que garanta uma boa QoE para o usuário. Tal solução integrada também deve incluir módulos adicionais que, por exemplo, dêem suporte para: (1) a



recepção e transmissão de fluxos de dados (conteúdo e controle) via redes diversas; (2) a sincronização dos fluxos recebidos e enviados; (3) a descoberta e o pareamento de dispositivos em redes locais; e (4) a distribuição da aplicação entre esses diversos dispositivos.

Nesse sentido de solução multimídia integrada, a principal contribuição do laboratório TeleMídia é o *middleware* Ginga [Gomes Soares et al. 2010], que hoje é o *middleware* padrão do sistema brasileiro de TV digital terrestre [ABNT 2016] e recomendação UIT-T para sistemas IPTV [ITU-T 2014]. O *middleware* Ginga é o resultado de anos de pesquisa em sistemas multimídia e envolveu o esforço de vários alunos de graduação, mestrado e doutorado que passaram pelo laboratório TeleMídia, além de diversas contribuições de outros grupos de pesquisa.

A Figura 3 apresenta a arquitetura do *middleware*. O Ginga é dividido em dois módulos principais: Common-Core e Ginga-NCL. O Common-Core é o módulo responsável por instanciar os *players* dos tipos de mídia suportados em uma determinada plataforma (TV digital terrestre, rádio digital ou IPTV), intermediar a comunicação com as diversas redes e dispositivos remotos, e gerenciar o ciclo de vida das aplicações recebidas. Já o Ginga-NCL é o exibidor NCL propriamente dito, isto é, o módulo responsável por apresentar as aplicações NCL.



**Figura 3. Arquitetura do *middleware* Ginga.**

Dando continuidade ao desenvolvimento das próximas gerações do *middleware* Ginga, atualmente o laboratório TeleMídia mantém o projeto Ginga 2.0. Esse projeto tem como objetivo principal aperfeiçoar a implementação de referência do Ginga e do exibidor Ginga-NCL, incorporando as novas funcionalidades previstas pela NCL 4.0, além de promover a evolução do *middleware* em diferentes plataformas de TV digital terrestre, rádio digital e IPTV. São objetivos específicos do projeto Ginga 2.0 integrar ao *middleware* suporte às seguintes funcionalidades:

1. Pré-busca inteligente de dados sob demanda (*pulled data*).

2. Pré-busca inteligente de dados sem solicitação (*pushed data*) em carrossel de objetos.
3. Negociação antecipada de QoS (*Quality of Service*) para dados sob demanda (*pulled data*).
4. Extensão do modelo de apresentação do formatador para escalonamento de aplicações iniciadas em qualquer ponto do tempo de sua duração.
5. Atualização dinâmica dos componentes do *middleware*.

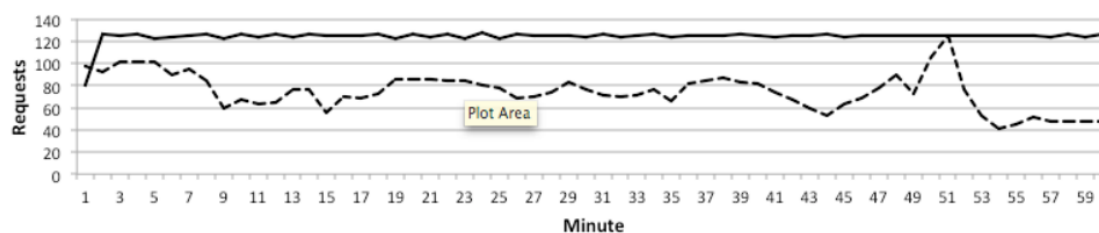
### ***Ginga IBB, BemTV e audio watermarking***

Outro tema recorrente nas pesquisas do laboratório TeleMídia, relacionado ao tópico de arquitetura e implementação de sistemas multimídia, é o desenvolvimento de *players* de objetos de mídia e de protocolos de transmissão ou sincronização que resolvam problemas específicos de um determinado domínio multimídia. Os exemplos mais recentes de projetos desse tipo desenvolvidos pelo laboratório são: (1) a integração de suporte à IBB (*Integrated Broadcast/Broadband*) ao padrão ABNT e à recomendação UIT-T do Ginga; (2) a biblioteca BemTV para balanceamento de carga via comunicação P2P (*peer-to-peer*) em fluxos de vídeo; e (3) um sistema para sincronismo de mídia em larga escala via *audio watermarking*.

Atualmente, serviços de TV Conectada (ou Smart TV) provêm a distribuição de conteúdo por redes de transmissão de TV (satélite, terrestre ou cabo) e internet (por exemplo, via Ethernet DSL, Ethernet no CATV ou LTE). Entretanto, eles não necessariamente oferecem serviços convergentes que fazem o uso conjunto de conteúdos de ambas as redes de distribuição. Os serviços IBB [ITU 2016] têm sido propostos com o objetivo de suportar uma convergência real entre as redes de transmissão de TV e a internet, e permitir, por exemplo, a vinculação do conteúdo oferecido por redes via satélite, terrestre ou cabo ao conteúdo da internet. Nesse sentido, o *middleware* Ginga também suporta serviços IBB, por meio de sinalizações no fluxo de transporte de TV que permitem a sincronização dos dados no fluxo de transmissão aos serviços de conteúdo da internet. Através desta sinalização é possível, por exemplo, iniciar o portal de conteúdo do radiodifusor do canal correspondente, o que permite que o espectador do canal acesse serviços de vídeo sob demanda relacionados ao conteúdo que está sendo transmitido.

O BemTV [Barbosa and Soares 2014] é uma biblioteca desenvolvida no laboratório TeleMídia para o balanceamento de carga na distribuição de fluxos HLS (*HTTP Live Streaming*). A biblioteca utiliza comunicação P2P para compartilhar *chunks* comuns entre os clientes de um mesmo vídeo e, dessa forma, diminuir o número de requisições à CDN (*Content Delivery Network*). Essa técnica é particularmente atrativa nos casos em que o vídeo é captado ao vivo, em alta qualidade, e precisa ser distribuído em tempo real a um grande número de clientes parcialmente sincronizados (por exemplo, partidas de futebol, concertos, etc.).

A Figura 4 a seguir apresenta o resultado de um experimento em que dois *players* requisitam *chunks* à CDN. O primeiro *player* utiliza apenas a CDN; o segundo utiliza o modelo híbrido CDN-P2P do BemTV. Enquanto o primeiro *player* (linha contínua) realizou 7457 requisições à CDN, o segundo (linha tracejada) realizou apenas 4482, ou seja, uma redução de 39,89% no total de requisições.



**Figura 4. Requisições à CDN e requisições aos *peers*** [Barbosa and Soares 2014].

Em algumas aplicações a sincronização—e não a distribuição—do conteúdo é que precisa ser feita em larga escala. Esse é o caso, por exemplo, de aplicações em que um relógio global precisa ser distribuído a um grupo de dispositivos confinados (próximos uns dos outros) mesmo na ausência de redes tradicionais. Uma forma de atender a esse requisito (assumindo que os dispositivos possuem microfone) é através de ondas sonoras via *audio watermarking*.

*Audio watermarking* é uma técnica utilizada para a adição de informações adicionais em fluxos de áudio. É comumente utilizada na indústria para incorporação de metadados em faixas musicais, por exemplo, título da música, autor, copyright, etc. Alguns trabalhos recentes propõem o uso dessa técnica para sincronizar conteúdo multimídia em ambientes controlados. Em [Arnold et al. 2014], por exemplo, o *audio watermarking* é aplicado ao som transmitido pela TV e é utilizado para sincronizar dispositivos móveis. Já em [Lemos et al. 2016] a técnica é utilizada para sincronização local de aplicações Web.

Diferente desses trabalhos, o laboratório TeleMídia tem pesquisado a aplicação de *audio watermarking* para sincronização de dispositivos em ambientes abertos, onde há uma grande aglomeração de pessoas e não há rede, ou a rede não é confiável. Cenários comuns desses ambientes são shows e eventos para multidões em ambientes abertos. São alvos dessa pesquisa a investigação da usabilidade da técnica (incômodo gerado e efeitos da exposição prolongada) e da sua viabilidade tecnológica quando considerada a transmissão de dados em altas frequências, isto é, frequências entre 18KHz e 20KHz.

### 7.2.3. Experiências imersivas

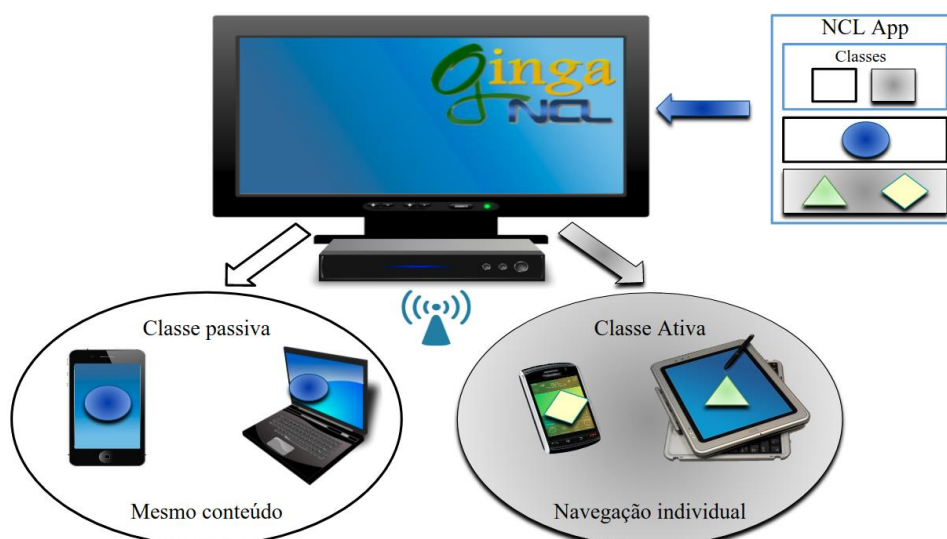
O objetivo desta linha de pesquisa é estender as representações de cenas (Seção 1.2.1) e exibidores multimídia (Seção 1.2.2) para possibilitarem uma maior imersão dos usuários durante a apresentação. Com esse objetivo, o laboratório TeleMídia vem realizando pesquisas sobre como aumentar a QoE do usuário através de novas formas de visualização e interação. Na parte de visualização, essas pesquisas consideram o uso de múltiplas telas de exibição e de displays 3D. Já na parte de interação, os estudos desenvolvidos consideram o uso de múltiplas modalidades de interação (por exemplo, reconhecimento de gestos e voz, e outras mídias sensoriais em geral.)

#### *Múltiplos dispositivos e telas secundárias*

O ambiente de TV é um ambiente de assistência coletiva, isto é, é comum que várias pessoas assistam o mesmo programa ao mesmo tempo. Nesse cenário, quando uma dos telespectadores interage com a TV, por exemplo, para responder um questionário interativo, a qualidade de experiência dos demais telespectadores (aqueles que querem

apenas continuar assistindo o programa principal) pode ser prejudicada. Uma solução para esse problema é delegar a interação à dispositivos secundários (ou de segunda tela) permitindo assim que cada telespectador tenha uma interação personalizada. Exemplos de dispositivos secundários são *smartphones*, *tablets* e *smartwatches*.

O *middleware* Ginga suporta aplicações com dispositivos secundários desde a sua primeira versão [Soares et al. 2009]. Para prover esse suporte, o Ginga utiliza um modelo hierárquico de distribuição de conteúdo e de controle entre os múltiplos dispositivos. Os dispositivos secundários registram-se como filhos do dispositivos base (em geral, a TV) e passam a fazer parte de uma classe de dispositivos, conforme ilustrado na Figura 5. Dessa forma, o dispositivo pai pode controlar o conteúdo que é exibido nos dispositivos filhos. NCL 3.0 define duas classes de dispositivos: *classe passiva*, em que todos os dispositivos da classe apresentam simultaneamente o mesmo conteúdo; e *classe ativa*, em que a navegação e conteúdo exibido é individualizada para cada dispositivo da classe.



**Figura 5. Modelo hierárquico de controle de dispositivos do Ginga.**

A Figura 6 abaixo apresenta exemplos de aplicações que utilizam múltiplos dispositivos secundários desenvolvidas em NCL 3.0 para o *middleware* Ginga.



**Figura 6. Exemplos de aplicações de segunda tela usando o Ginga.**

Diversos tipos de aplicações de videocolaboração podem ser implementadas em NCL 3.0 utilizando o suporte da linguagem à múltiplos dispositivos. Isso é especialmente interessante no caso de dispositivos da classe ativa. Por exemplo, durante

a apresentação de um vídeo na tela principal, os usuários da aplicação podem colaborar utilizando seus smartphones para realizar uma determinada tarefa, ou podem entreter-se por meio de jogos colaborativos relacionados com o vídeo principal (como os exemplos da Figura 6).

Como discutido na Seção 1.2.1, a próxima versão da linguagem NCL permitirá, além das classes passivas e ativas, que novas classes de dispositivos customizadas sejam criadas pelos autores. Uma proposta inicial de como é possível a definição de novas classes de dispositivos (ainda não padronizada) é discutida em [Batista 2013].

### ***Displays 3D e vídeos 3D***

Outras tecnologias com grande potencial para aumentar a imersão e melhorar a colaboração entre usuários em sistemas de videocolaboração são aquelas relacionadas à *displays* e vídeos 3D. Um exemplo de sistemas de videocolaboração com esse tipo de suporte são os sistemas de videoconferência 3D que tiram proveito de tecnologias de *displays* e vídeos 3D estereoscópicos, e que vem sendo pesquisados há bastante tempo.

*Displays* 3D estereoscópicos são *displays* que fornecem ao usuário dicas binoculares de profundidade através de óculos especiais. Apesar dessa tecnologia existir há bastante tempo, nos últimos anos a produção desse tipo de *display* cresceu consideravelmente impulsionada pela produção massiva de filmes 3D. *Displays* 3D estereoscópicos, entretanto, possuem algumas limitações importantes, por exemplo, a necessidade do uso de óculos especiais e a possibilidade de provocar náuseas em algumas pessoas.

Com o objetivo de solucionar algumas dessas limitações, novas tecnologias de *displays* 3D vêm sendo desenvolvidas. Exemplos dessas tecnologias incluem *head-mounted displays* (HMD), *displays* autoestereoscópicos e *displays* baseados em *light field*. Os dois últimos fornecem dicas binoculares aos usuários sem a necessidade de óculos especiais.

São exemplos de *head-mounted displays* o Google Cardboard (Figura 7.a) e o Microsoft HoloLens (Figura 7.b). Já o HoloVizio (Figura 7.c) é um exemplo de *display light field*. A Figura 7.d apresenta, esquematicamente, o funcionamento de um *display light field*: múltiplos projetores são usados para recriar um padrão de raios de luz de tal forma que o espectador possa ver uma imagem 3D sem a necessidade de óculos especiais e com paralaxe de movimento, isto é, o espectador percebe uma imagem diferente na medida em que movimenta a sua cabeça [Lawton 2011].



(a)

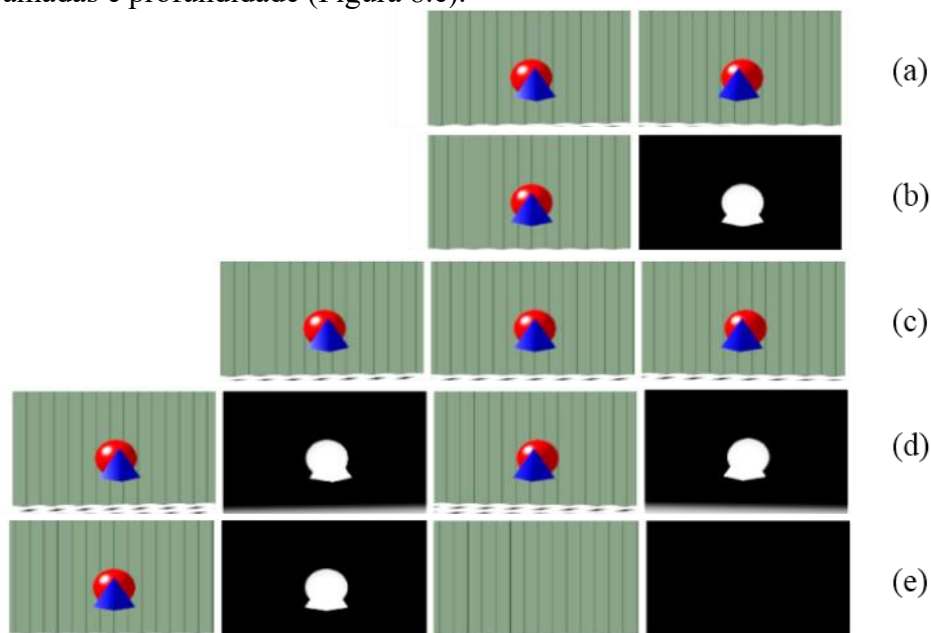


(b)



**Figura 7. Exemplos de displays 3D: (a) Google Cardboard, (b) Microsoft Hololens, (c) HoloVizio; e (d) esquema de funcionamento um *display light field*.**

Simultaneamente, para dar suporte à essas novas tecnologias de *displays* 3D e possibilitar o uso eficiente das redes de distribuição, novas representações de vídeo 3D também têm sido propostas. São exemplos dessas novas representações os formatos de vídeo estereoscópico convencional (Figura 8.a), vídeo com profundidade (Figura 8.b), vídeo multivista (Figura 8.c), vídeo multivista com profundidade (Figura 8.d), e vídeo com camadas e profundidade (Figura 8.e).



**Figura 8. Diferentes representações de vídeo 3D.**

Desde o início da década de 2010, o laboratório TeleMídia tem aplicado algumas dessas tecnologias de *displays* 3D e vídeos 3D ao contexto de aplicações multimídia interativas. Entre os trabalhos desenvolvidos estão a renderização em tempo real de vídeos com profundidade [Azevedo et al. 2014] e o suporte à autoria e execução de aplicações multimídia baseadas em profundidade [Azevedo 2015; Azevedo and Lima 2016; Azevedo and Soares 2013].

No que se refere a videocolaboração, quando comparadas aos sistemas baseados em vídeo estereoscópico convencionais, as tecnologias de vídeo 3D baseadas em profundidade (como as discutidas em [Azevedo et al. 2014]) tem as seguintes



vantagens: (1) utilizam menos banda, uma vez que um quadro de profundidade armazena menos informação do que um quadro de cor; (2) permitem ao cliente adaptar a profundidade às características de diferentes *displays* estereoscópicos e preferências do usuário; e (3) permitem ao cliente gerar vistas adicionais da cena, suportando assim *displays* autoestereoscópicos.

A inclusão de cenas interativas em ambientes de videoconferência 3D também é um tema que vem sendo pesquisado no laboratório. Além de possibilitar a reprodução de cenas de forma mais fiel ao mundo real e permitir cenários de realidade aumentada, onde objetos virtuais podem ser inseridos em vídeos 3D capturados, outra oportunidade interessante ao integrar-se cenas interativas em ambientes de videoconferência 3D é a possibilidade do uso da profundidade real para ressaltar características da interface com o usuário [Azevedo 2015].

### ***Interações multimodais e mídias sensoriais***

Recentes avanços em tecnologias de reconhecimento de sinais e o uso de novos dispositivos como sensores e atuadores vêm impulsionando o desenvolvimento de interfaces de usuário não-tradicionais, como, por exemplo, as interfaces multimodais. Interfaces multimodais são caracterizadas pelo uso (possivelmente simultâneo) de múltiplas modalidades dos sentidos humanos e por poderem combinar múltiplos dados de entrada (e.g., fala, caneta, toque, gesto, olhar e movimentos de cabeça e corpo) e saída (e.g., sintetização de voz e efeitos sensoriais) [Turk 2014]. Comparada às tecnologias atuais de mouse, teclado e *displays*, as tecnologias de interface multimodal permitem o desenvolvimento de interfaces com usuários mais naturais para a linguagem e comportamento humano.

Até recentemente, entretanto, linguagens multimídia como NCL e HTML, não suportavam interações multimodais. Tradicionalmente, essas linguagens privilegiam a criação de aplicações que usam elementos audiovisuais e que adotam um paradigma de interação WIMP (*Windows, Icons, Menus, Pointers*), pressupondo o uso de dispositivos de interações tradicionais como teclado, mouse e *touchscreens*.

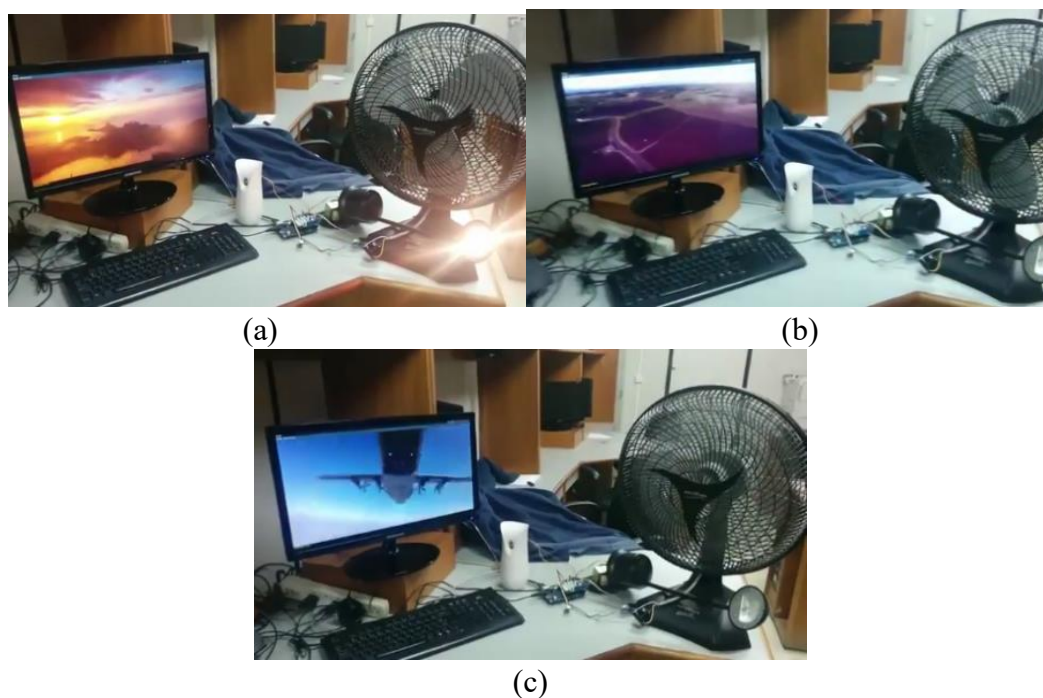
Relacionado ao tema de interações multimodais, o laboratório TeleMídia mantém uma frente de pesquisa que visa incorporar extensões a linguagens multimídia que permitam: (1) o uso de novas modalidades de entrada, como reconhecimento de voz e de gestos através de descrições de reconhecimento como SRGS e GML; (2) o uso de novas modalidades de saída, como fala sintetizada e atuadores através de descrições como o SSML (*Speech Synthesis Markup Language*) [W3C 2004] —linguagem para especificação de aplicações de sintetização de voz— e SEDL (*Sensory Effect Description Language*) [ISO 2016] —linguagem parte da suíte de padrões MPEG-V para a especificação de aplicações que descrevem efeitos sensoriais como iluminação, ventilação, vibração, etc.; e (3) a especificação da sincronização entre essas diferentes modalidades de entrada e saída.

Como resultado preliminar dessas pesquisas, propostas de extensões multimodais para NCL 3.0 são detalhadas em [Guedes, A. L. et al. 2016]. A Figura 9 apresenta uma versão alternativa da aplicação com múltiplos dispositivos da Seção 1.2.2 (Figura 5), agora com suporte à interação por gestos.



**Figura 9. Exemplo de aplicação multimodal com reconhecimento de gestos.**

Relacionado especificamente a novos tipos de atuadores, em [Guedes, Á. et al. 2016], é discutido um exemplo de aplicação que apresenta um vídeo de forma sincronizada a efeitos de iluminação, cheiro e vento (Figura 10). Enriquecer aplicações multimídia com o uso de objetos que estimulam outros sentidos de forma sincronizada pode contribuir para aumentar a sensação de imersão dos usuários dessas aplicações. Tais aplicações são algumas vezes chamadas de aplicações multi-sensoriais ou *MulseMedia* [Sulema 2016],



**Figura 10. Aplicação com efeitos sensoriais de luz (a), cheiro (b) e vento (c).**

### **7.3. Visão de futuro em videocolaboração**

Considerando as tecnologias e pesquisas discutidas nas seções anteriores, esta seção apresenta quatro possíveis cenários de evolução para aplicações de vídeo interativo e de videocolaboração.



### 7.3.1. Videocolaboração em ambientes domésticos

A comunicação intermediada por computador tem se tornado uma parte indispensável das interações diárias da maioria das pessoas. Em ambientes empresariais e de telemedicina, sistemas de videoconferência de alta qualidade e telepresença, por exemplo, já são uma realidade. Tais sistemas, em geral, são implantados utilizando-se redes privadas gerenciadas e hardware dedicado. No entanto, avanços atuais em infraestrutura de rede e poder computacional vêm possibilitando o desenvolvimento de novas aplicações de videocolaboração também em ambientes domésticos.

Cenários de videocolaboração usando redes locais—e.g. aplicações de segunda tela em que os usuários estão em uma mesma rede local [Buchner et al. 2014]—já são possíveis de serem implantados com boa qualidade de serviço, incluindo aqueles que demandam sincronismo fino entre os objetos a serem apresentados nos diversos dispositivos. Entretanto, a manutenção desse sincronismo em aplicações em que usuários estão geograficamente distribuídos e usando redes não-gerenciadas ainda é um problema para a tecnologia atual.

Aplicações de *Social TV* são exemplos de aplicações de videocolaboração em que o suporte ao sincronismo fino mesmo entre usuários geograficamente distribuídos é indispensável. Em aplicações de Social TV, usuários geograficamente distantes assistem ao mesmo conteúdo ao vivo enquanto interagem uns com os outros em tempo real. O YouTube, por exemplo, já suporta transmissões de eventos ao vivo e permite que os espectadores dessas transmissões troquem mensagens enquanto assistem a um vídeo. Porém, as mensagens e o vídeo do YouTube não são sincronizadas, e não há qualquer garantia de que todos os clientes estão sincronizados (assistindo exatamente o mesmo quadro do vídeo). Possibilidades de que os usuários possam controlar a transmissão junto com seus pares, por exemplo, avançando e retrocedendo o vídeo para que possam comentar sobre um determinado evento da transmissão, também são exemplos de desenvolvimentos futuros em videocolaboração e Social TV.

Outro exemplo de aplicação de videocolaboração que pode se beneficiar do suporte ao sincronismo fino entre usuários geograficamente distribuídos são os ambientes de aprendizagem online (*e-learning*). Atualmente plataformas como Coursera<sup>2</sup>, MIT OpenCourseWare<sup>3</sup> e Udacity<sup>4</sup> oferecem serviços de vídeo interativo em salas de aula virtuais em que professores e alunos interagem por meio de fóruns (interação tipicamente assíncrona). Esses serviços poderiam ser aperfeiçoados através de técnicas que garantem que professores e alunos compartilhem a mesma experiência remotamente, de forma que as mensagens trocadas e o conteúdo exibido estejam sempre sincronizados em todos os clientes, o que não ocorre hoje.

### 7.3.2. Videoconferência 3D e telepresença

Há pouca dúvida de que um dos caminhos na evolução dos sistemas de videocolaboração é a utilização cada vez maior de tecnologias de *displays* e vídeo 3D, as quais estão constantemente evoluindo. Esses avanços tem como objetivo promover

---

<sup>2</sup> <https://www.coursera.org/>

<sup>3</sup> <https://ocw.mit.edu/>

<sup>4</sup> <https://www.udacity.com/>

uma experiência cada vez mais próxima da comunicação face-a-face que temos diariamente.

Os vários avanços dos últimos anos em tecnologias de captura, codificação, transmissão, renderização e *displays* 3D de fato já tornou viável o uso de tecnologias estereoscópicas tradicionais em sistemas de videoconferência de baixo custo. Mesmo assim, tecnologias estereoscópicas tradicionais ainda não são utilizados em larga escala em videocolaboração, principalmente, quando consideramos cenários domésticos. De fato, o principal motivo de ainda não termos o uso rotineiro dessas tecnologias parece estar relacionado aos a baixa QoE dos *displays* estereoscópicos tradicionais (vide Seção 1.2.3.). Nesse sentido, em um curto e médio prazo possivelmente veremos aplicações de videoconferência baseadas principalmente em *head-mounted displays* e *displays* baseados em *light field*, os quais resolvem alguns dos problemas dos *displays* estereoscópicos tradicionais.

*Head-mounted displays* possibilitam uma sensação de imersão bem maior do que os *displays* estereoscópicos tradicionais, o que é bastante interessante para aplicações de videocolaboração 3D e telepresença. De fato, atualmente, existe bastante interesse no uso de *head-mounted displays* e vídeos 360/VR em videocolaboração. Muito desse interesse está relacionado à disponibilidade de vários novos tipos de *head-mounted displays* de baixo custo e com boa qualidade. Além disso, a crescente disponibilização de *smartphones* também tem permitido o reúso de tais dispositivos como telas em dispositivos VR de baixo custo, e.g., Google Cardboard.

Atualmente, pesquisadores e grandes *players* da indústria (Google e Facebook) já estudam e desenvolvem tecnologias relacionadas à captura, transmissão e renderização de vídeos 360/VR. Por exemplo, várias tecnologias para codificação e transmissão de vídeos 360 (e.g., por meio de técnicas adaptativas [Hosseini and Swaminathan 2016]) têm sido desenvolvidas e podem ser facilmente adaptadas para sistemas de videoconferência. Essas tecnologias, principalmente aquelas baseadas em Web, têm potencial para se tornarem produtos de videocolaboração e irão poderão, em breve, estar disponíveis ao usuário final.

Com relação aos *displays* baseados em *light field*, a principal vantagem desses *displays* é que é possível ter uma visão estereoscópica e com paralaxe de movimento sem a necessidade de vestir nenhum dispositivo (e.g. óculos). Embora tais *displays* já estejam disponível em pequena escala, e.g. HoloVizio [Balogh et al. 2007], eles ainda são caros e requerem muito mais dados do que os *head mounted displays*. De qualquer forma, há um crescente interesse de pesquisa em codificação e compressão tanto de representações para *light fields* [Perra 2016; Zare et al. 2017] quando de representações para *displays* múltiplas vistas com profundidade [Purica et al. 2016; Zhang et al. 2017] que, em um médio prazo, podem viabilizar o uso dessa tecnologia também em aplicações de videocolaboração.

### 7.3.2. Videocolaboração com experiências multi-sensoriais

Atualmente, alguns cinemas, também chamados de cinemas 4D, buscam oferecer uma sensação realista para usuários utilizando, além de vídeos e *displays* 3D, também efeitos sensoriais adicionais, tais como vibração de assentos, iluminação e cheiro. Tais experiências, entretanto, ainda estão restritas a teatros especializados, com dispositivos específicos e que não possuem a capacidade de transmissão de conteúdo ao vivo.

Baseado nesses exemplos hoje já existentes, é possível vislumbrar que essas novas experiências sensoriais também possuem grande potencial de aumentar a imersão em sistemas de videocolaboração ao permitir ao público remoto uma experiência mais próxima à de estar no local do evento. Exemplos de aplicações que podem se beneficiar dessa maior imersão são as transmissões ao vivo de eventos esportivos e artísticos.

A reprodução realista de um local em outro pode ser alcançada com a apresentação coordenada do conteúdo audiovisual e de objetos reais, que podem ser controlados por meio de sensores e atuadores. Enquanto os sensores capturam características do ambiente, os atuadores reproduzem essas características, tentando assim reproduzir a experiência sensorial do local de origem no local de destino. Exemplos de atuadores que podem ser integrados em uma aplicação de videoconferência são: áudio direcional, vibração, efeitos luminosos e vento.

Entretanto, o uso de forma efetiva desses efeitos em ambientes de videocolaboração, ainda depende de estudos mais detalhados de como os usuários percebem e em quais circunstâncias tais efeitos sensoriais adicionais podem de fato contribuir para aumentar a qualidade de experiência dos usuários. Adicionalmente, efeitos sensoriais como aqueles baseados em cheiro e paladar possuem requisitos de sincronismo que ainda estamos longe de entender completamente e é difícil acreditar que poderão ser utilizados em ambientes reais em um curto período de tempo.

#### **7.3.4. Aplicações multimídia sensíveis à semântica do conteúdo**

Recentes avanços nas áreas de inteligência artificial e visão computacional buscam reconhecer automaticamente o conteúdo semântico de vídeos. Algumas abordagens utilizam aprendizagem de máquina para detectar características visuais de baixo nível no vídeo [Laptev et al. 2008; Ye et al. 2015] enquanto outras utilizam modelos semânticos (ontologias) para descrever conceitos de alto nível [Liu et al. 2013; Ye et al. 2015]. Apesar desses avanços, o desenvolvimento de aplicações multimídia ainda foca na captura e apresentação de conteúdo audiovisual e ignora a semântica desse conteúdo.

Aplicações multimídia sensíveis à semântica do conteúdo são aplicações que conseguem reconhecer os conceitos que estão sendo apresentados e disparar eventos em decorrência disso. Por exemplo, ao reconhecer um monumento histórico no vídeo que está sendo apresentado, o *player* de um portal de vídeos poderia apresentar textos, links ou vídeos relacionados à esse monumento.

Os principais desafios relacionados ao desenvolvimento desse tipo de aplicação são, primeiro, como fazer a aplicação entender o que é o conceito “monumento histórico”? E como fazê-la reconhecer o objeto “monumento histórico” no conteúdo no vídeo? A primeira questão é explorada pela comunidade de Web semântica, que busca definir vocabulários e ontologias que representam conceitos e relações semânticas entre eles. Já a segunda questão, é estudada pela área de visão computacional que pesquisa técnicas de reconhecimento de padrões para classificação automática de conteúdo.

Outra possibilidade interessante é desenvolvimento de aplicações que, além da estrutura e do conteúdo audiovisual, considerem a representação e a extração em tempo real da semântica do conteúdo apresentado ou do conteúdo gerado pelos usuários. Por exemplo, em sessões de videoconferência de grupos de trabalho ou de ensino a distância, o sistema poderia reconhecer os conceitos que estão sendo discutidos e exibir

automaticamente informações adicionais sobre esses conceitos ou conceitos relacionados.

#### **7.4. Conclusão**

Desde a sua fundação o laboratório TeleMídia vem atuando em pesquisas em sistemas multimídia e hipermídia, focando-se especialmente em problemas de representação de cenas multimídia interativas e no projeto e implementação de arquiteturas para sistemas multimídia.

Relacionado à representação de cenas multimídia interativas, as pesquisas do laboratório resultaram no modelo conceitual NCM, na sua linguagem NCL e em várias ferramentas de autoria. Atualmente, diversos trabalhos do laboratório estão focados na evolução tanto do modelo NCM quanto da linguagem NCL, e no desenvolvimento de novas ferramentas e aprimoramento das já existentes.

Relacionado à arquiteturas de sistemas multimídia, as pesquisas do laboratório estão voltadas principalmente para o acesso convergente e sincronizado de diferentes canais de distribuição (*Integrated Broadband Broadcast*), e para cenários que envolvam experiências imersivas. Esses cenários consideram o uso de experiências envolvendo múltiplos dispositivos e o uso de novas modalidades de mídia e interação, e.g. via vídeos 3D, efeitos sensoriais e interação multimodal.

No futuro, o laboratório pretende pesquisar cenários de sistemas de multimídia e videocolaboração que ofereçam uma experiência de comunicação cada vez mais próxima àquela da experiência face-a-face, ou mesmo cenários que superem a experiência real. Esses novos cenários se apoiam na visão de *Sincronismo das Coisas* [Soares 2014a, 2014b], que considera que o requisito de sincronismo pode ocorrer em diferentes níveis em um sistema multimídia, por exemplo, entre fluxos mídia oriundos de diferentes canais distribuição (IBB), entre dispositivos geograficamente distribuídos, entre interações de usuário via novas modalidades entrada (voz e gestos), etc. Além disso, os novos cenários consideram sistemas que se adaptam dinamicamente dependendo da semântica do conteúdo que está sendo apresentado e da semântica da entrada fornecida pelos usuários do sistema.

## Biografia dos autores

**Sérgio Colcher** é professor do quadro principal do Departamento de Informática (DI) da PUC-Rio desde 2001 e coordenador do laboratório TeleMídia. Obteve os títulos de Engenheiro de Computação (1991), Mestre em Ciências em Informática (1993) e Doutor em Ciências em Informática (1999), todos pela PUC-Rio, além do Pós-Doutorado (2003) no ISIMA (Institute Supérieur D'Informatique et de Modelisation des Applications — Université Blaise Pascal, Clermont Ferrand, França). Trabalhou no Centro Científico da IBM-Rio e na divisão de desenvolvimento de hardware da COBRA (Computadores Brasileiros S/A). Foi também professor dos cursos de MBA em Gerência de Telecomunicações e MBA em e-Business da Fundação Getúlio Vargas. É autor do livro “Redes de Computadores: das LANs, MANs e WANs às Redes ATM” (Ed. Campus/Elsevier 1995), livro indicado ao prêmio Jabuti de 1996, e do livro “VoIP: Voz sobre IP” (Ed. Campus/Elsevier 2005). Suas áreas de interesse incluem redes de computadores, análise de desempenho de sistemas computacionais, sistemas multimídia/hipermídia e sistemas de TV digital.

**Álan L. V. Guedes** é aluno de doutorado na PUC-Rio e pesquisador associado ao laboratório TeleMídia. Ele recebeu o título de bacharel em Ciência da Computação (2009) e de mestre em Informática (2012) pela UFPB, onde atuou como pesquisador associado no laboratório Lavid. Seus interesses de pesquisas incluem as áreas de sistemas multimídia e multimodal, e a autoria de aplicações para estes sistemas.

**Roberto Gerson de A. Azevedo** é pesquisador associado ao laboratório TeleMídia. Possui doutorado (2015) e mestrado (2010) em Informática pela PUC-Rio e é bacharel em Ciência da Computação pela UFMA (2008). Seus interesses de pesquisa incluem: representação e autoria de cenas multimídia interativas; e representação, codificação, transmissão e renderização de vídeos 3D.

**Guilherme F. Lima** é pesquisador associado ao laboratório TeleMídia. Ele possui doutorado (2015) e mestrado (2011) em informática, e o título de bacharel (2009) em Sistemas de Informação, todos pela PUC-Rio. Seus principais interesses de pesquisa são linguagens de programação e modelos para sincronização multimídia, em particular, na interseção entre linguagens síncrona e multimídia.

**Rodrigo C. M. Santos** é aluno de doutorado na PUC-Rio e pesquisador associado ao laboratório TeleMídia. Ele recebeu o título de bacharel em Ciência da Computação (2010) e mestre em Ciência da Computação (2012) pela UFMA, onde trabalhou como pesquisador associado ao laboratório LAWS. Seus interesses de pesquisas incluem as áreas de linguagens de programação para multimídia, modelos de sincronismo distribuído e sistemas multimídia distribuídos.

**Antonio José G. Busson** é aluno de doutorado na PUC-Rio e pesquisador associado ao laboratório TeleMídia. Ele recebeu o título de bacharel em Ciência da Computação (2013) e de mestre em Ciência da Computação (2015) pela UFMA, onde atuou como pesquisador associado ao laboratório LAWS. Seus interesses de pesquisa incluem: sistemas multimídia/hipermídia, modelos de hiperdocumentos, reconhecimento de padrões e sistemas de TV Digital.

## Bibliografia

- ABNT (2016). ABNT NBR 15606-2:2016 EN Digital terrestrial television - Data coding and transmission specification for digital broadcasting Part 2: Ginga-NCL for fixed and mobile receivers - XML application language for application coding. <http://www.abntcatalogo.com.br/norma.aspx?ID=361857>.
- Antonacci, M. J., Saade, D. C. M., Rodrigues, R. F. and Soares, L. F. G. (jun 2000). NCL: Uma Linguagem Declarativa para Especificação de Documentos Hipermedia na Web. In *VI Simpósio Brasileiro de Sistemas Multimídia e Hipermedia - SBMídia2000*.
- Araújo, E. C. and Soares, L. F. G. (2014). Designing iDTV Applications through Interactive Storyboards. In *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web - WebMedia '14*. . <http://dx.doi.org/10.1145/2664551.2664572>.
- Arnold, M., Chen, X.-M., Baum, P., Gries, U. and Doerr, G. (2014). A Phase-Based Audio Watermarking System Robust to Acoustic Path Propagation. *IEEE Transactions on Information Forensics and Security*, v. 9, n. 3, p. 411–425.
- Azevedo, R. G. A., Araújo, E. C., Lima, B., Soares, L. F. G. and Moreno, M. F. (2012). Composer: meeting non-functional aspects of hypermedia authoring environment. *Multimedia tools and applications*, v. 70, n. 2, p. 1199–1228.
- Azevedo, R. G. A., De Salles Soares Neto, C., Teixeira, M. M., Santos, R. C. M. and Gomes, T. A. (2011). Textual authoring of interactive digital TV applications. In *Proceedings of the 9th international interactive conference on Interactive television - EuroITV '11*. . <http://dx.doi.org/10.1145/2000119.2000169>.
- Azevedo, R. G. de A. (2015). Supporting Multimedia Applications in Stereoscopic and Depth-based 3D Video Systems. PUC-Rio.
- Azevedo, R. G. de A., Ismério, F., Raposo, A. B. and Soares, L. F. G. (2014). Real-Time Depth-Image-Based Rendering for 3DTV Using OpenCL. *Lecture Notes in Computer Science*. p. 97–106.
- Azevedo, R. G. de A. and Lima, G. F. (2016). A graphics composition architecture for multimedia applications based on layered-depth-image. In *2016 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*. . <http://dx.doi.org/10.1109/3dtv.2016.7548882>.
- Azevedo, R. G. de A. and Soares, L. F. G. (2013). NCL+depth: Estendendo NCL para Displays 3D Estereo/Autoestereoscópicos. In *Proceedings of the 19th Brazilian symposium on Multimedia and the web - WebMedia '13*. . <http://dx.doi.org/10.1145/2526188.2526203>.
- Balogh, T., Kovács, P. T. and Megyesi, Z. (2007). HoloVizio 3D Display System. In

- Proceedings of the ImmersCom*. .  
<http://dx.doi.org/10.4108/icst.immerscom2007.2145>.
- Barbosa, F. R. N. and Soares, L. F. G. (2014). Towards the Application of WebRTC Peer-to-Peer to Scale Live Video Streaming over the Internet. In *Anais do IX Workshop de Redes P2P, Dinâmicas, Sociais e Orientadas a Conteúdo - Wp2p+*.
- Batista, C. E. C. F. (2013). GINGA-MD: UMA PLATAFORMA PARA SUPORTE À EXECUÇÃO DE APLICAÇÕES HIPERMÍDIA MULTI-DISPOSITIVO BASEADA EM NCL. Pontifícia Universidade Católica do Rio de Janeiro.
- Bernal, I. and Mejía, D. (2015). Building a Basic Hardware and Software Infrastructure for Developing Ginga-NCL Interactive Applications. *Applications and Usability of Interactive TV*. Springer, Cham. p. 74–89.
- Buchner, K., Lissermann, R. and Holmquist, L. E. (2014). Interaction techniques for co-located collaborative TV. In *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems - CHI EA '14*. .  
<http://dx.doi.org/10.1145/2559206.2581257>.
- Bulterman, D. C. and Rutledge, L. (2016). *SMIL 3.0: Flexible Multimedia for Web, Mobile Devices and Daisy Talking Books*. Springer.
- Casanova, M. A., Tucherman, L., Lima, M. J. D., et al. (1991). The nested context model for hyperdocuments. In *Proceedings of the third annual ACM conference on Hypertext - HYPERTEXT '91*. . <http://dx.doi.org/10.1145/122974.122993>.
- Cesar, P. and Geerts, D. (2011). Past, present, and future of social TV: A categorization. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)*. .  
<http://dx.doi.org/10.1109/ccnc.2011.5766487>.
- Fechine, Y. (2017). Social TV: A contribution towards defining the concept. *Revista Contracampo*, v. 36, n. 1.
- Ferreira, T. P. and Souza Filho, G. L. (2014). Uma extensão da visão estrutural do NCL Composer para integração de código imperativo.
- Gomes Soares, L., Soares, L. G., Moreno, M., Neto, C. S. S. and Moreno, M. (2010). Ginga-NCL: Declarative middleware for multimedia IPTV services. *IEEE Communications Magazine*, v. 48, n. 6, p. 74–81.
- Guedes, Á., Cunha, M., Fuks, H. and Colcher, S. (2016). Using NCL to Synchronize Media Objects, Sensors and Actuators. In *In: Workshop Internacional de Sincronismo das Coisas (WSoT), 1, 2016, Teresina. Anais do XXII Simpósio Brasileiro de Sistemas Multimídia e Web. Porto Alegre: Sociedade Brasileira de Computação, 2016. v. 2. , WebMedia '16*.
- Guedes, A. L., Azevedo, R. G. A. and Barbosa, S. D. J. (2016). Extending multimedia

- languages to support multimodal user interactions. *Multimedia tools and applications*, v. 76, n. 4, p. 5691–5720.
- Guedes, Á. L., Azevedo, R. G. de A., Colcher, S. and Barbosa, S. D. J. (2016). Extending NCL to Support Multiuser and Multimodal Interactions. *Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web - Webmedia '16*,
- Hosseini, M. and Swaminathan, V. (dec 2016). Adaptive 360 VR Video Streaming: Divide and Conquer. In *2016 IEEE International Symposium on Multimedia (ISM)*. . IEEE.
- ISO (2016). ISO/IEC 23005-3:2016 - Information technology -- Media context and control -- Part 3: Sensory information. [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=65396](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=65396), [accessed on Sep 5].
- ITU (2016). Report BT.2267-6. <http://www.itu.int:80/en/publications/ITU-R/Pages/publications.aspx>.
- ITU-T (2014). ITU Recommendation H.761: Nested context language (NCL) and Ginga-NCL for IPTV services. <http://handle.itu.int/11.1002/1000/12237>.
- Jansen, J., Frantzis, M. and Cesar, P. (2015). Multimedia Document Structure for Distributed Theatre. In *Proceedings of the 2015 ACM Symposium on Document Engineering - DocEng '15*. . <http://dx.doi.org/10.1145/2682571.2797087>.
- Laptev, I., Marszalek, M., Schmid, C. and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*. . <http://dx.doi.org/10.1109/cvpr.2008.4587756>.
- Lemos, V. S., Ferreira, R. F., Costa Segundo, R. M., Costalonga, L. L. and Santos, C. A. S. (2016). Local Synchronization of Web Applications with Audio Markings. In *Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web - Webmedia '16*. . <http://dx.doi.org/10.1145/2976796.2976853>.
- Liu, J., Cheng, H. and Javed, O. Et al (2013). SRI-Sarnoff AURORA System. In *TRECVID 2013 Multimedia Event Detection and Recounting*.
- Luiz Fernando Gomes and Barbosa, S. D. J. (2009). *Programando em NCL 3.0: desenvolvimento de aplicações para middleware Ginga: TV digital e Web*.
- Meixner, B. (6 mar 2017). Hypervideos and Interactive Multimedia Presentations. *ACM Computing Surveys*, v. 50, n. 1, p. 1–34.
- Mejía, D., Bernal, I. and Becerra, F. (1 feb 2014). Plugin para Composer NCL y Aplicación Interactiva para TV Digital Orientada a Educación Superior. *Revista* , v. 33, n. 1.



- Perra, C. (2016). Light field image compression based on preprocessing and high efficiency coding. In *2016 24th Telecommunications Forum (TELFOR)*. . <http://dx.doi.org/10.1109/telfor.2016.7818930>.
- Purica, A. I., Mora, E. G., Pesquet-Popescu, B., Cagnazzo, M. and Ionescu, B. (2016). Multiview Plus Depth Video Coding With Temporal Prediction View Synthesis. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 26, n. 2, p. 360–374.
- Sant’Anna, F., Cerqueira, R. and Soares, L. F. G. (2008). NCLua. In *Proceedings of the 14th Brazilian Symposium on Multimedia and the Web - WebMedia '08*. . <http://dx.doi.org/10.1145/1666091.1666107>.
- Santos, R. C. M., Neto, J. R. C., De Salles Soares Neto, C. and Teixeira, M. M. (2013). Incremental validation of NCL hypermedia documents. *Journal of the Brazilian Computer Society*, v. 19, n. 3, p. 235–256.
- Soares, L. F. G. (2014a). Sincronismo das Coisas não é apenas expressão da moda. <http://video.rnp.br/portal/video.action?idItem=23850>, [accessed on Sep 6].
- Soares, L. F. G. (2014b). Sincronismo das Coisas Não É Apenas Expressão da Moda. In *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*. . <http://www.lbd.dcc.ufmg.br/colecoes/wvcwebmedia/2014/009.pdf>, [accessed on Sep 6].
- Soares, L. F. G., Costa, R. M. R., Moreno, M. F. and Moreno, M. F. (2009). Multiple exhibition devices in DTV systems. In *Proceedings of the seventeen ACM international conference on Multimedia - MM '09*. . <http://dx.doi.org/10.1145/1631272.1631312>.
- Soares, L. F. G., Rodrigues, R. F., Costa, R. R. and Moreno, M. F. (2006). Nested context language 3.0: Part 9--ncl live editing commands. *Monografias em Ciência da Computação do Departamento de Informática, PUC-Rio*, v. 6, p. 36.
- Soares Neto, C. de S., De Salles Soares Neto, C., Soares, L. F. G. and De Souza, C. S. (2012). TAL—Template Authoring Language. *Journal of the Brazilian Computer Society*, v. 18, n. 3, p. 185–199.
- Sulema, Y. (2016). Mulsemmedia vs. Multimedia: State of the art and future trends. In *2016 International Conference on Systems, Signals and Image Processing (IWSSIP)*. . <http://dx.doi.org/10.1109/iwssip.2016.7502696>.
- Turk, M. (jan 2014). Multimodal interaction: A review. *Pattern recognition letters*, v. 36, p. 189–195.
- Van Deventer, M. O., Stokking, H., Hammond, M., Le Feuvre, J. and Cesar, P. (mar 2016). Standards for multi-stream and multi-device media synchronization. *IEEE Communications Magazine*, v. 54, n. 3, p. 16–21.

- W3C (2004). Speech Synthesis Markup Language (SSML) Version 1.0. <https://www.w3.org/TR/speech-synthesis/>, [accessed on Sep 5].
- Ye, G., Li, Y., Xu, H., Liu, D. and Chang, S.-F. (2015). EventNet: A Large Scale Structured Concept Library for Complex Event Detection in Video. In *Proceedings of the 23rd ACM international conference on Multimedia - MM '15*. . ACM Press.
- Zare, A., Balogh, T., Bregovic', R. and Gotchev, A. (2017). Architectures and Codecs for Real-Time Light Field Streaming. *Electronic Imaging*, v. 2017, n. 5, p. 54–66.
- Zhang, Q., Huang, K., Wang, X., Jiang, B. and Gan, Y. (2017). Efficient multiview video plus depth coding for 3D-HEVC based on complexity classification of the treeblock. *Journal of Real-Time Image Processing*,