UNIVERSIDADE DE SÃO PAULO

ESCOLA DE ENGENHARIA DE SÃO CARLOS

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

E DE COMPUTAÇÃO

SEL0630 - Aplicações de Microprocessadores II

R3V - *Remote 3D Viewer*

Autor(es): Davi Diório Mendes 7546989

Henrique Alberto Rusa 7593714

Professor: Evandro Luis Linhari Rodrigues

2015

Resumo

Texto em um parágrafo apenas - deve conter "tudo"resumidamente (introdução, método(s), resultados e conclusões), de tal forma que seja possível compreender a proposta e o que foi alcançado.

Palavras-Chave: palavra1, palavra2, palavra3, palavra4, palavra5.

Sumário

Capítulo 1

Introdução

Realmente introduz o leitor indicando quais são as direções do trabalho ? apresenta o tema e o objeto do trabalho e contém as Referências do Estado da arte (quem está fazendo e em que nível os trabalhos da área estão hoje) [1].

Outra referência para a bibliografia [2].

Segundo [3] há uma sequência lógica para a redação da monografia como apresenta em [4].

Referência para a figura 1.1.

Figura 1.1: Logo da EESC.

1.1 Objetivos

Objetivos do trabalho.

1.2 Motivação (opcional)

Descrever a motivação do trabalho.

1.3 Justificativas/relevância(opcional)

Justificativa do trabalho.

1.4 Organização do Trabalho(opcional)

Este trabalho está distribuído em XXX capítulos, incluindo esta introdução, dispostos conforme a descrição que segue:

Capítulo 2: Descreve .....

Capítulo 3: Discorre sobre .....

Capítulo 4: Apresenta .....

Capítulo 2

Embasamento Teórico

Na fase de projeto do sistema R3V proposto foram elencados conhecimentos e fundamentos necessários para o desenvolvimento do mesmo. Com isso em mente, pode-se discursar melhor sobre os tópicos mais relevantes do sistema, permitindo que o leitor aprofunde-se devidamente para que, ao final, os objetivos do projeto sejam melhor discutidos e compreendidos.
Neste capítulo pretende-se analisar os conceitos de estereoscopia, sistemas embarcados, sistemas distribuídos e *dead reckoning*.

2.1 Estereoscopia

A simulação de imagens em três dimensões (visão espacial) é associada ao conceito da estereoscopia. Para isso, o cérebro humano necessita adiquirir duas imagens, cada qual com um leve deslocamento lateral (angular), e com estas calcular a profundidade dos objetos.
Existem também técnicas de percepção do espaço tri-dimensional que avaliam sombras, sobreposição de objetos numa cena e vários outros parâmetros, possibilitando verificar a disposição dos elementos no espaço.
Entretanto, existe uma diferença fundamental entre percepção e simulação do espaço tri-dimensioanl para o usuário, apresentadas a seguir.

- **Percepção Tri-dimensonal:** O usuário consegue perceber a disposição dos elementos numa imagem (profundidade relativa, distância relativa entre objetos) somente a partir da análise de uma imagem que possui indicativos como sombra, iluminação e outros.

- **Simulação Tri-dimensonal:** O usuário tem a sensação de que os objetos possuem dimensões reais, com tamanho e profundidade bem definidos; a distância entre elementos é percebida,

entretanto não mais pela sombra, mas pela própria projeção da simulação em si.

Portanto, enquanto o usuário pode perceber a terceira dimensão em valores relativos (objetos mais próximos ou mais afastados, maiores ou menores), a simulação do espaço tri-dimensioanl é realizada pela sobreposição de imagens com mesmo ponto focal mas deslocadas levemente uma da outra, possibilitando a reconstrução, pelo cérebro, da profundidade dos elementos.

2.2 Sistemas Embarcados

Segundo White (2011, p.1) a definição de sistemas embarcados varia de pessoas para pessoas. Para alguém acostumado a lidar com servidores, programação para dispositivos móveis pode ser considerada como um desenvolvimento embarcado. Por outro lado, para aquele acostumado a trabalhar com microcontroladores de 8-bits, qualquer coisa com um sistema operacional não parece muito embarcado. Mas ela enuncia a definição: "um sistema embarcado é um sistema computadorizado construído propositadamente para sua aplicação" citar aqui.

Desta forma podemos realçar a diferença entre sistemas computacionais e sistemas embarcados. Enquanto sistemas computacionas são desenvolvidos para atuar em computadores de propósito geral, sistemas embarcados são desenvolvidos de forma a serem embarcados em *hardware* com propósitos específicos, visando a aplicação.

2.3 Sistemas Distribuídos

Sistemas distribuídos é algo de difícil definição. Cada autor utiliza uma definição diferente, embora todas remetam que há computadores, ou outro dispositivo com capacidade de processamento, trabalhando em conjunto. Vamos tomar por definição: "Um sistema distribuído é aquele no qual os componentes localizados em computadores interligados em rede se comunicam e coordenam suas ações apenas passando mensagens." (COULOURIS, 2007, p. 15).

2.4 *WebServices REST*

Hoje a internet não é mais simplesmente uma ferramenta de difusão de informação. Diversos serviços são prestados através dos *Web Services*. Normalmente envolvem um processo burocrático rigoroso, como requisições SOAP ou outros tipos de mensagem, conforme o protocolo seguido. A idéia de um *WebService REST* é de prover serviços diretamente sobre o protocolo HTTP, sem a necessidade de adicionar uma camada de protocolo de serviço. Serviços *REST* estão se popularizando cada vez mais, dada a baixa complexidade do sistema e fácil integração por parte dos usuários.

2.5 *Dead Reckoning*

A tecnologia provê diversos sistemas de posicionamento por sensores, *beacons* (referências fixas no espaço), equivalência entre mapas e outros. Entretanto, pode-se distinguir duas metodologias utilizadas: posicionamento relativo e absoluto.

- **Absoluto:** Se vale de técnicas de sensoreamento com referências fixas, previamente implementadas, o que adiciona um custo muito alto para cosntrução e manutenção destes sistemas espalhados pelo terreno em questão.
- **Relativo:** Propõe uma formulação mais elegante, se valendo de sensores e parâmetros intrínsecos da implementação. Com isso, pode-se estimar a posição atual do elemento de acordo com movimentações perceptíveis ao sistema.

Dead reckoning é uma implementação de posicionamento relativo que permite o programador avaliar os diversos sensores para se estimar a localização entre os períodos de tempos avaliados.
Note que as duas implementações de posicionamento possuem suas vantagens e desvantagens, sendo que uma estimativa de localização (posição relativa) adiciona erros às medidas, enquanto que a outra possui as referências extremamente precisas (posição absoluta).

2.6 Rotação dos Eixos

A decomposição dos movimentos de rotação de um usuário utilizando um óculos de virtualização é realizada com movimentos angulares, descritas pelos valores *pitch*, *yaw* e *roll*.
A figura 2.1 apresenta a orientação das rotações descritas anteriormente.

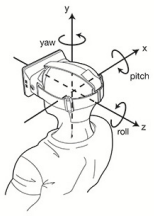

Figura 2.1: Representação dos movimentos de visão de um usuário (visto em: https://s3.amazonaws.com/static.oculus.com/website/2013/05/oculus_head_model.jpg)

Capítulo 3

Materiais e Métodos

Neste seção irá se discursar sobre os materiais e métodos utilizados no desenvolvimento do projeto *R3V*. Os materiais englobam todos os dispositivos físicos e produtos lógicos (bibliotecas de código, etc) utilizados no desenvolvimento do projeto proposto. Os métodos apresentam como e por meio de qual técnica alguns processos do sistema foram desenvolvidos, abstraindo o produto e focando na maneira com que foi implementado.

3.1 Materiais

A seguir são apresentados os materiais relevantes para o projeto proposto (*R3V*), sendo eles: placa de desenvolvimento Intel Galileo, *smarthphone Android*, IPCam (câmera controlada por rede) e Flask (servidor python).

3.1.1 Intel Galileo

A placa de desenvolvimento Intel® Galileo (figura 3.1), projetada e vendida pela Intel®, foi criada com base no processador Intel® Quark SoC X1000 de aplicação com o intuito de ser compatível com os *shields* de Arduino.

O processador provê para o usuário final uma arquitetura de 32 bits e *clock* de 400 MHz, conectores Ethernet 10/100, PCI-Express *mini-card*, USB 2.0, USB cliente (usado para programação) e botões de *reboot* e *reset*.

Como apresentado anteriormente, a placa possui diversas interfaces de conexão para viabilizar a comunicação com um computador, outro Arduino e até outros microcontroladores.

A Galileo suporta programação através da IDE do Arduino como também a utilização de um sistema operacional embarcado instalado em seu *hardware*.

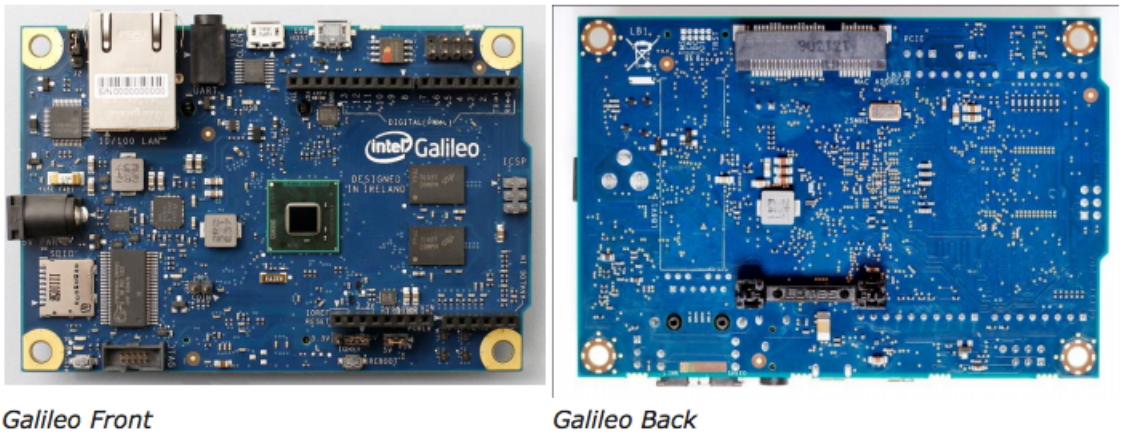

Figura 3.1: Imagem da placa de desenvolvimento Galileo - visão frontal (*Galileo Front*), visão de trás (*Galileo Back*) (visto em: http://www.intel.com/content/www/us/en/embedded/products/galileo/galileo-g1-datasheet.html)

A sua aplicação dentro do projeto desenvolvido é prover um *webservice* REST através do *framework* Flask.

3.1.2 Smartphone Android

Atualmente, os *smartphones* Android vem sendo uma ótima plataforma para desenvolvimento. Já integrado com diversos sensores e dispositivos de entrada e saída; como acelerômetro, GPS, câmera, microfone, sistema de som, tela, entre outros; além de disponibilizar uma sólida ferramenta de desenvolvimento, a Android SDK.

Dos *hardwares* providos pelo *smartphone* foi utilizado na aplicação os sensores inerciais, a fim de obter os angulos de euler da orientação do dispositivo; a tela, para a exibição da interface gráfica e a conexão *wifi*, para comunicação com a Intel Galileo. A forma como estes recursos foram utilizados será abordada na exposição da aplicação Android.

3.1.3 IPCam

Para o *streaming* de video, uma câmera IP foi utilizada afim de disponibilizar um serviço em rede e desacoplar o dispositivo do servidor (placa de desenvolvimento Intel® Galileo). Este *hardware* provê, através de um protocolo *http*, um serviço que executa diversas operações através de *scripts* CGI.
As intertfaces CGI da câmera apresentam três níveis diferenciados de permissão, sendo eles: visitante, operador e administrador. A funcionalidade utilizada pelo projeto somente requisitou acesso de operador (usuário e senha fornecidos pelo professor).

Além de serviços de video, a câmera em questão possui *scripts* para monitoramento, detecção de movimento, movimentação da mesma e diversos outros.
O projeto R3V utilizou-se de dois *scripts*, chamados: *decoder_control.cgi* e *videostream.cgi*.
O *script videostream.cgi* realiza a captura e envio do video através da rede para o cliente, enquanto que o *decoder_control.cgi* controla a movimentação da câmera em sua base. Abaixo (tabela 3.1) se destaca os comandos de movimentação que foram utilizados no projeto.

Tabela 3.1: Comandos utilizados no projeto R3V da câmera IPCam FOSCAM

Comando	Descrição
0	Move a câmera para baixo
2	Move a câmera para cima
4	Move a câmera para a esquerda
6	Move a câmera para a direita
25	Coloca a câmera no seu centro (implementação própria)
90	Move a câmera para a diagonal esquerda inferior
91	Move a câmera para a diagonal direita inferior
92	Move a câmera para a digonal esquerda superior
93	Move a câmera para a diagonal direita superior

Para mais informações sobre o funcionamento da câmera utilizada consulte o manual do fabricante: [5].

3.1.4 Flask

"O Flask é um microframework para python, baseado no *Werkzeug*, *Jinja2* e boas intenções" **referência para página do flask**. Com o *Flask* podemos facilmente criar um *backend* para processar as requisições à Intel Galileo, implementando assim um *WebService* REST.

Para utilizarmos o Flask, atrelamos um método python a uma determinada URL. Assim conseguimos recuperar dados transmitidos através da requisição HTTP, processá-los e retornarmos uma resposta. Através do Flask, o R3V provê três serviços: *streaming* de vídeo de uma câmera IP, movimentação desta câmera e *dead reckoning* do movimento da câmera. A seguir abordamos melhor o provisionamento de cada um destes serviços.

Através da URL <*endereço_da_intel_galileo*>/camstream/ provemos o serviço de *streaming* de vídeo. Este endereço retorna um *stream* MJPEG ouseja, uma sequência de imagens JPEG. Este serviço recupera as imagens da câmera pelo mesmo serviço de MJPEG. Desta forma ele atua somente como um *proxy*, encapsulando o serviço de *stream* de vídeo da própria câmera IP.

Através da URL <*endereço_da_intel_galileo*>/camposition/cam_step/?move=<direção> provemos o serviço de movimentação da câmera IP em passos de cinco graus. No parâmatro direção deve-se informar o código da direção que se deve atuar. As direções são mapeadas conforme a tabela 3.2.

Tabela 3.2: XABLAU

Código	Direção
0	abaixo
2	acima
4	esquerda
6	direita
90	diagonal esquerda abaixo
91	diagonal direita abaixo
92	diagonal esquerda acima
93	diagonal direita acima

Através da URL <*endereço_da_intel_galileo*>/camposition/set_zero/ inicializamos o sistema de *dead reckoning* do movimento da câmera, tomando a posição atual da câmera como zero graus de rotação em torno do eixo *x* e zero graus de rotação em torno do eixo *y*.

Através da URL <*endereço_da_intel_galileo*>/camposition/?pitch=<x>&yaw=<y> provemos o serviço de movimentação da câmera com *dead reckoning* do movimento. No parâmetro *x* deve-se atribuir qual a posição angular desejada em torno do eixo *x*. No parâmetro *y* deve-se atribuir qual a posição angular desejada em torno do eixo *y*. O serviço de movimentação por *dead reckoning* só aceita posições angulares para *pitch* entre 80 e -30 graus e posições angulares para *yaw* entre 100 e -100 graus.

Desta forma temos a Intel Galileo provendo os serviços de *streaming*, *movimentação* e *dead reckoning* da posição de uma câmera IP.

3.2 Métodos

No desenvolvimento de um projeto várias técnicas são utilizadas para implementar diversos componentes necessários para o correto funcionamento do sistema, tendo em vista eficiência e qualidade no produto final.

Nesta seção serão apresentados todos os métodos, implementações e comportamentos dos componentes desenvolvidos para se construir o sistema R3V. Com isso, irá se discursar um pouco sobre

streaming de video (MJPEG), mapeamento de movimento (*dead reckoning*) e o servidor desenvolvido.

3.2.1 MJPEG

3.2.2 *Dead Reckoning*

Uma das estruturas fundamentais do comportamento geral do sistema R3V se compõe do mapeamento e controle dos movimentos do usuário, afim de proporcionar uma fluidez na movimentação da câmera no lado da IPCam.

Utilizando os conceitos de *dead reckoning* (posicionamento de acordo com leitura de sensores e estimativa de movimento), uma transformação matemática foi aplicada aos resultados do sensor inercial (acelerômetro) adiquiridos através do aplicativo do *smartphone*, resultando em um discretização do espaço de visão do usuário. Isso se deve ao fato de que a IPCam possui somente movimentos bem definidos e não responde a movimentos fora de seus comandos pré-definidos.

Através do SDK do *Cardboard* da Google® foi possível adiquirir os movimentos angulares denominados *pitch* e *yaw*, que correspondem a movimentos em torno de um eixo bem definido (eixo x e y respectivamente).

A transformação realizou a conversão do movimento para um valor da superfície de uma esfera (quadrante), por meio de uma função trigonométrica. Assim

$$arctan2(x,y) = \begin{cases} arctan(\frac{y}{x}) & x > 0 \\ arctan(\frac{y}{x}) + \pi & x < 0 \; and \; y \geq 0 \\ arctan(\frac{y}{x}) - \pi & x < 0 \; and \; y < 0 \\ \frac{\pi}{2} & x = 0 \; and \; y > 0 \\ -\frac{\pi}{2} & x = 0 \; and \; y < 0 \\ undefined & x = 0 \; and \; y = 0 \end{cases}$$

O resultado de $arctan2(x,y)$ retorna um valor entre $[-\pi,\pi]$. Este valor apresenta o quadrante o qual a visão do usuário está olhando no momento. Portanto, foi possível discretizar o espaço do campo de visão em segmentos, onde o mapeamento da visão possibilita, através dos comandos padrões da câmera IPCam, simular um movimento natural do usuário no dispositivo ligado em rede.

Uma ressalva a se fazer a este método é que o sistema necessita armazenar o valor da última posição para que se possa realizar o cálculo do deslocamento angular da câmera. Com isso, foi possível se rastrear e mapear o campo de visão com o movimento do usuário.

3.2.3 Applicação Cliente

A android SDK provê meios de criar interfaces gráficas, desenvolver aplicativos internacionalizáveis (em diversas línguas), aplicativos com processamento paralelo, comunicação interprocesso, conexão com servidores HTTP, e muitas outras funcionalidades. Neste projeto nos interessa a criação de uma interface gráfica para utilizar o *smartphone* no Google Cardboard, o acesso aos sensores inerciais (acelerômetro, giroscópio e magnetômetro) e a comunicação com servidores HTTP.

Para a interface gráfico, o necessário é podermos exibir uma imagem em cada metade da tela, com o celular em orientação paisagem. Ambas as imagens deve ser a mesma, de forma a dar o aspecto tridimensional quando no *cardboard*. Para nos beneficiarmos de algumas facilidades da *Cardboard SDK*, discutidas adiante, foi necessário manter a interface de renderizador gráfico da *cardboard*, mas uma nova interface foi sobreposta sobre a padrão. No código fonte.

Capítulo 4

Resultados e Discussões

Aqui se mostra o que o trabalho permitiu produzir, e às vezes o que pode ser comparado com outros trabalhos - aqui ficam claras se as propostas do trabalho são relevantes ou não, pois devem permitir a discussão do trabalho.

Deve responder: Os resultados estão claros em bom número (nem muito nem pouco) que permitam avaliar realmente a proposta e o que foi produzido.

4.1 *Streaming* de Imagem

4.2 *Dead Reckoning*

4.3 *Frontend*

Capítulo 5

Conclusão ou Conclusões

"Fecha"com os objetivos? (respondem aos objetivos?)

Valorizam (ou não) o trabalho realizado. Normalmente é uma parte do trabalho "um pouco desprezada", pois o autor já está "cansado....".

Mas é aqui o lugar que se pode medir se o trabalho tem ou não valor.

Trabalhos futuros

É uma orientação sobre as possibilidades de continuação do desenvolvimento do trabalho.

Referências Bibliográficas

[1] Autor da referência 1. Título da referência 1, 2007.

[2] Google. `http://www.google.com.br/`, Acesso em: 04 de dezembro de 2014.

[3] E.L.L. Rodrigues. Dicas, cuidados e orientações para a elaboração de texto para tcc, 2015.

[4] Enzo Bertini Vieira e Lara Bertini Vieira. Sistema autônomo de vigilância baseado em dados biológicos com registro de dados na nuvem via smartphone, 2014.

[5] Shenzhen Foscam Technology. *IP Camera CGI*. 37 pp.

Apêndice A

Complementos importantes do texto

Observe as diretrizes de redação no site do Depto.

http://www.sel.eesc.usp.br/informatica/graduacao/tcc/tcc_-_diretrizes_EESC_v_2010.pdf).

Aqui são colocadas as informações de autoria própria, porém entendidas como complemento da informação contida no corpo do trabalho. São colocadas aqui para não "carregar"demais o texto.

Atenção para as **Referências Bibliográficas**: todas as referências citadas no texto. Observar as Diretrizes, pois lá estão os formatos corretos de citação.

Outras observações **IMPORTANTES** (leia isso com atenção)

NUNCA copie texto de outro autor sem a devida forma de citação (ver em diretrizes); a cópia configura plágio! Com a Internet e/ou outras ferramentas dedicadas, é muito fácil identificar se houve cópia de texto.

⇒ figura que não é de sua autoria deve conter a fonte;

⇒ no texto, toda primeira vez que aparecer algum protocolo, procedimento, nome técnico, sigla, abreviatura, etc, além de explicar o que é, é necessário citar a referência. Exemplo: ...um giroscópio (referência) é um tipo de sensor...

⇒ capriche nas figuras (uma figura bem composta quase não precisa de texto para explicá-la);

⇒ procure manter uma "uniformidade de notação"para o texto todo;

⇒ não tenha medo de citar os trabalhos de outros autores (isso é imprescindível);

⇒ evite muitas referências de sites, pois são voláteis;

- ⇒ NÃO USE O WIKIPEDIA COMO REFERÊNCIA;
- ⇒ todas as palavras escritas em inglês (ou em outras línguas) devem estar em itálico;
- ⇒ todas as figuras e tabelas devem ser referenciadas no texto;
- ⇒ todas as obras citadas nas referências bibliográficas devem estar citadas no texto;
- ⇒ códigos de programas devem estar em Apêndices, pois servem para comprovar o desenvolvimento e facilitar a reprodução do trabalho;

Apêndice B

Apresentação do Trabalho

Como tem-se até 30 minutos para fazer a apresentação deve-se dimensionar a quantidade de slides para isso. Cada um tem seu "timming"com relação à quantidade de informação versus tempo disponível para apresentação.

Os slides devem ser sempre muito mais visuais que textuais, ou seja, não se deve colocar frases e "ficar lendo"as mesmas. Os slides devem apresentar uma forma "clean"para que sirva apenas de guia para a apresentação do trabalho.

Não carregue de texto os slides...

Anexo I

Anexo 1

Material que não é de sua autoria, mas que são importantes e devem fazer parte da monografia para auxiliar e esclarecer o leitor;

Anexo II

Anexo 2

Texto do Anexo 2.