

1

Introdução

Diversos ramos da engenharia e da geociências fazem uso de aplicações de visualização 3D. Estas permitem que os profissionais daquelas áreas possam planejar melhor suas ações através da análise de objetos 3D que representem estruturas do mundo real. Essa etapa tem se tornado cada vez mais importante, pois possibilita reduzir custos e maximizar a produção.

Na área de exploração e produção de petróleo isso não é diferente. Atualmente existem diversas aplicações de visualização 3D, cada uma delas com o enfoque em uma etapa diferente do processo de extração de petróleo. Por exemplo, um determinado aplicativo pode ser capaz de mostrar modelos CAD de engenharia, como plataformas de petróleo ou navios, mas não permite a visualização de reservatórios ou poços de petróleo simultaneamente. Ainda não existe uma aplicação que permita a visualização integrada desses modelos.

Entretanto, há uma necessidade crescente de ferramentas desse tipo. Com elas seria possível ter uma visão gerencial integrada dos modelos usados nas diversas etapas do projeto, permitindo a realização de um planejamento mais preciso das ações que devem ser tomadas.

Para atender a essa demanda, o TecGraf e a Petrobras começaram a trabalhar conjuntamente no projeto SiVIEP (Sistema de Visualização Integrado de Exploração e Produção). Seu objetivo é criar uma aplicação para visualizar de forma integrada os diferentes modelos 3D presentes no processo de extração e produção de petróleo. Apesar de ainda estar em estágio de desenvolvimento, a aplicação já é capaz de mostrar vários tipos de modelos e possui algumas ferramentas de navegação (Figura 1.1).

Tal integração é necessária e representa uma grande vantagem em relação aos aplicativos presentes no mercado, mas ela acaba criando novos problemas que geralmente não existiam quando cada modelo era visualizado separadamente. O primeiro, e mais evidente, tem relação com o desempenho: agora uma quantidade muito maior de geometria deve ser carregada na placa gráfica e isso pode impactar negativamente na taxa de quadros por segundo que a aplicação é capaz de exibir. Nesse sentido, técnicas de aceleração em renderização devem ser implementadas.

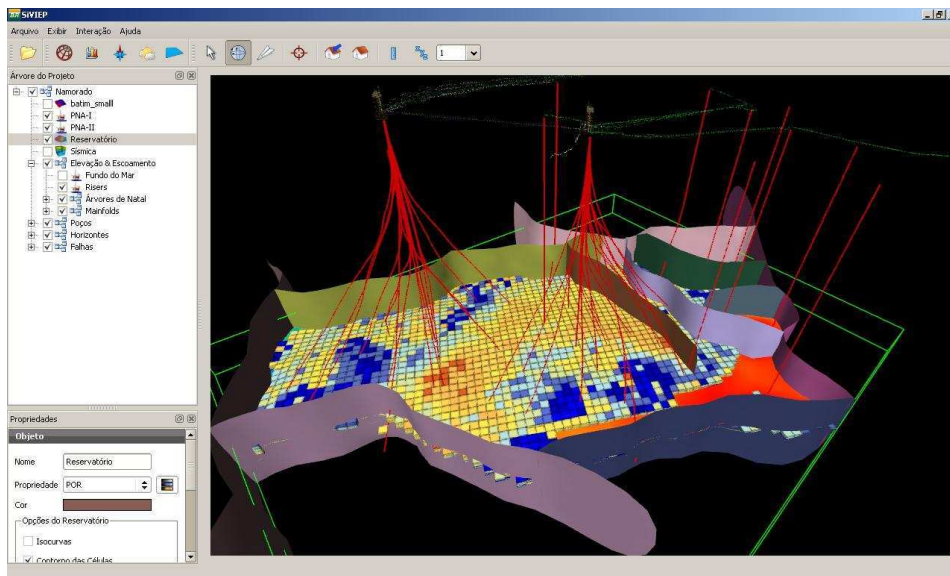


Figura 1.1: SiVIEP: visualização de um campo de exploração de petróleo.

Outro problema não menos importante tem relação com a *navegação* nesse novo tipo de ambiente virtual. Por navegação, entende-se como o *processo cognitivo de determinar e seguir um caminho, baseado no conhecimento e informações existentes no ambiente* [15].

No mundo real, a navegação é algo natural ao ser humano. As pessoas precisam a todo momento se deslocar de um ponto a outro e na maioria dos casos não há dificuldade em determinar como isso pode ser feito. Estudos mostram que os seres humanos são capazes de formar mapas cognitivos de seus ambientes para uso posterior na tarefa de navegação [6; 11; 13]. Diversas informações presentes no ambiente podem ser usadas para a construção desse mapa. Por exemplo, uma pessoa pode se basear no formato geométrico das construções ou mesmo em suas cores. Em um grande número de cidades, as ruas têm placas com os nomes das mesmas. Em alguns casos, é possível usar até mesmo o cheiro de um local como pista de sua localização.

Quando se trata de ambientes virtuais 3D, entretanto, as coisas podem ser bem diferentes e mais complicadas. O ato de navegar deixa de ser algo natural e, para alguns usuários, pode até mesmo resultar em experiências desagradáveis e frustrantes. Não é incomum ouvir frases do tipo ‘onde eu estou?’ ou ‘o objeto sumiu da tela’.

As causas para esses problemas são várias e têm sido objeto de estudo de muitos pesquisadores ao longo dos últimos anos. Por exemplo, em [15] é identificado um problema que os autores chamam de *desert fog*: condição na qual o ambiente imediatamente próximo não contém informações suficientes que permitam tomar decisões de navegação. Esse tipo de situação pode ocorrer em qualquer tipo de ambiente virtual, mas é mais comum em *ambientes*

multiescala (AMEs).

Ambientes multiescala são ambientes onde a informação pode existir em diferentes níveis de detalhe. Em outras palavras, em AMEs é possível visualizar simultaneamente objetos com diferentes escalas, indo desde um parafuso até um campo de petróleo com dezenas de quilômetros de extensão.

A integração de modelos 3D presente no SiVIEP proporciona justamente um ambiente virtual com essa característica. Por exemplo, um reservatório de petróleo é muito maior do que uma plataforma de extração, que por sua vez é bem maior que uma peça de maquinário localizada em seu interior. Como consequência, o SiVIEP acaba herdando alguns dos problemas de navegação inerentes a AMEs.

A característica multiescala, entretanto, não é a única fonte de problemas. Quando mal projetadas, as ferramentas de navegação podem acabar criando situações confusas. No artigo *Safe 3D Navigation* [9], por exemplo, os autores chegam a algumas conclusões interessantes sobre o comportamento de algumas pessoas ao usarem aplicações 3D:

- Nem sempre elas conseguem compreender como uma determinada ferramenta de navegação funciona ou, pior, a entendem incorretamente e acabam esperando um outro tipo de comportamento ao usá-la.
- Geralmente são necessárias mais do que uma única ferramenta para explorar o mundo virtual. Entretanto, um número considerável de usuários usa apenas uma ferramenta para todos os fins.
- Uma vez perdidas na cena, as pessoas geralmente tentam resolver a situação usando a mesma ferramenta de quando se originou o problema. O resultado disso é elas acabam não conseguindo reverter a situação, e pioram o problema.

Nesse sentido, o objetivo deste trabalho é propor e implementar soluções para os problemas de navegação que foram identificados durante o desenvolvimento do SiVIEP. Esses serão abordados com mais detalhes nas próximas seções, assim como também será feita uma descrição do SiVIEP e de suas ferramentas de navegação.

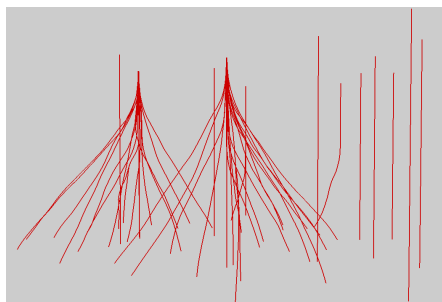
1.1 SiVIEP

A navegação em um determinado local depende fortemente do tipo do ambiente que esse possui [6]. Por exemplo, as decisões de uma pessoa serão provavelmente bem diferentes se ela estiver se deslocando no meio de um deserto ao invés de uma cidade.

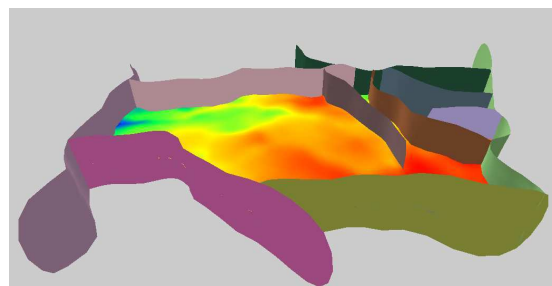
Por esse motivo, é conveniente mostrar e caracterizar alguns dos modelos que o SiVIEP é capaz de visualizar. Basicamente, existem dois tipos principais:

1. Modelos de geociências

- **Poços:** são perfurações na superfície terrestre utilizadas para produzir petróleo ou gás natural. São representados por vários segmentos de linhas que formam a trajetória final do poço (Figura 1.2(a)).
- **Superfície geológica:** ou falha geológica, é uma superfície em um volume de rocha onde é possível observar que houve deslocamento relativo dos blocos paralelos à falha. São representados por superfícies formadas pela composição de triângulos (Figura 1.2(b)).



1.2(a): Poços



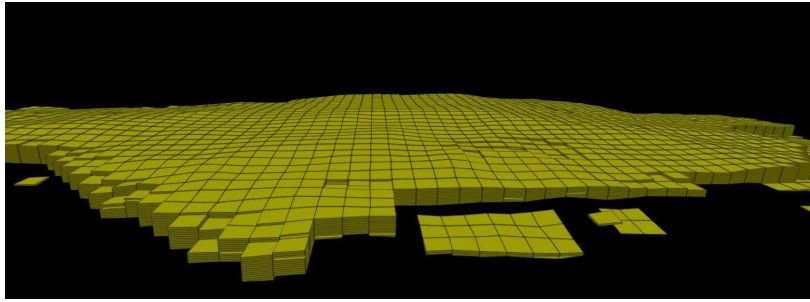
1.2(b): Superfícies Geológicas

Figura 1.2: Modelos de geociências

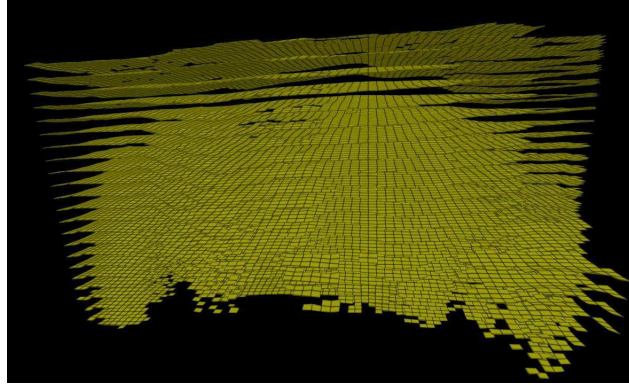
- **Reservatório:** corpo subterrâneo de rocha com porosidade e permeabilidade suficientes para armazenar e transmitir fluidos. Pode ser representado por uma malha discreta de células em formato de hexaedros, dispostos em grades e geralmente com mais de uma camada [5]. É possível estabelecer uma separação entre as camadas para permitir uma melhor inspeção (Figura 1.3).

2. Modelos de engenharia

- **Plataforma:** é representada por modelos CAD e é um dos objetos mais complexos (Figura 1.4(a)). É possível inspecionar esses modelos tanto externamente quanto internamente.
- **Árvore-de-natal:** sistema posicionado no fundo do mar, composto por válvulas conectadas ao poço e à unidade de produção na superfície (Figura 1.4(b)). Estas válvulas permitem o fluxo de produção de petróleo e gás, do poço para a superfície, assim como a injeção de líquido e gás da superfície para o poço.
- **Manifold:** estrutura metálica apoiada no fundo do mar, que acomoda válvulas e acessórios que permitem que este seja conectado às



1.3(a): Sem separação de camadas



1.3(b): Com separação de camadas

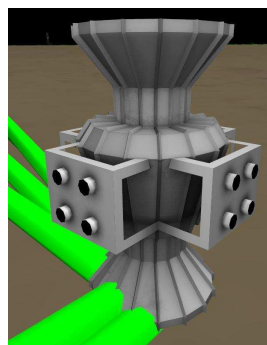
Figura 1.3: Modelo de reservatório

árvore-de-natal e a outros sistemas de produção, indo de tubulações a risers (Figura 1.4(c)).

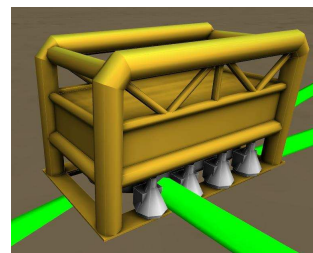
- **Riser:** conjunto de tubos flexíveis que conectam a unidade de produção e a árvore-de-natal e/ou manifold de produção, permitindo a injeção de gás ou a retirada de óleo e gás.



1.4(a): Plataforma



1.4(b): Árvore de natal



1.4(c): Manifold

Figura 1.4: Modelos de Engenharia

As definições acima foram retiradas de [29] e [25].

Para permitir que os usuários naveguem por esses modelos, o SiVIEP conta com duas ferramentas principais de navegação: *voar* e *examinar*.

A ferramenta *voar*, como o próprio nome diz, permite que a câmera ‘voe’ pela cena, em uma determinada velocidade, e o usuário pode dessa forma explorar livremente o ambiente. As setas direcionais do teclado transladam a câmera, enquanto o mouse é usado para orientar a direção do movimento.

A ferramenta *examinar* tem como objetivo possibilitar que o usuário examine um determinado objeto. Movimentos de arraste usando o botão esquerdo do mouse rotacionam a câmera em torno de um centro de rotação que pode ser fixado pelo usuário. A operação de *pan*, que permite que o usuário desloque o ambiente lateralmente como se esse fosse uma imagem 2D, é realizada através do ‘drag’ com o botão central do mouse. O ‘drag’ vertical/horizontal com botão direito do mouse tem efeito *zoom-in/zoom-out* respectivamente. O botão de scroll do mouse, quando presente, também é usado para operações de *zoom*.

1.2

Problemas de Navegação

Nas seções a seguir, são discutidos alguns problemas referentes a navegação em ambientes virtuais. Esses foram identificados a partir da observação do comportamento de diferentes usuários do SiVIEP.

1.2.1

Velocidade de Navegação

Níveis diferentes de escala exigem que a câmera tenha velocidades de navegação diferentes, compatíveis com a escala em que ela se encontra.

Para entender porque isso é necessário, suponha que um usuário esteja explorando o interior de uma plataforma usando a ferramenta *voar*. Nesse caso pode ser conveniente que a velocidade de navegação da câmera seja compatível com a de um ser humano andando, por exemplo. Mas se em um dado momento for necessário navegar até outra plataforma, que está distante 5 km da primeira, a velocidade deve ser reajustada. Caso contrário o tempo necessário para chegar ao destino seria muito grande. Conclui-se então que é necessária alguma maneira de se ajustar a velocidade de navegação.

Como tentativa de encontrar uma solução para esse problema, pode-se tentar observar como essa situação poderia ser resolvida no mundo real. Provavelmente, para ir de uma plataforma a outra se usaria algum tipo de veículo, por exemplo um barco motorizado. Com este seria possível acelerar e desacelerar, permitindo que a velocidade fosse ajustada quando fosse necessário.

Nesse exemplo, o controle está nas mãos do usuário. Logo, uma solução imediata para o problema no mundo virtual seria fornecer também alguma

maneira que o permitisse controlar a velocidade de navegação. O scroll do mouse pode ser usado para esse fim: dependendo do sentido de rolamento, a velocidade da ferramenta *voar* é aumentada ou diminuída.

Apesar de funcionar bem na maioria da vezes quando em situações reais, deixar o controle a cargo do usuário se revelou ineficaz no mundo virtual. Ao observar algumas pessoas enquanto usavam o SiVIEP, notou-se que um comportamento acontecia com certa frequência: ao perceberem que estavam em uma escala maior, elas aumentavam exageradamente a velocidade. Uma vez que objeto de interesse ainda estava longe, mesmo que a velocidade fosse dobrada, o usuário não tinha imediatamente a percepção de estar se aproximando do destino. Para compensar, os usuários então aumentavam ainda mais a velocidade, até sentirem o efeito esperado. Quando se aproximavam do objeto, entretanto, a velocidade estava alta demais e aquele acabava sendo ultrapassado. Isso acontecia mesmo com as pessoas que tentavam diminuir a velocidade usando o scroll do mouse, pois não conseguiam realizar a operação a tempo. Como resultado, o que se via era um processo iterativo, onde os usuários iam e voltavam até atingirem o destino. Esse problema também é descrito em [18], onde o autor relata a dificuldade das pessoas em controlar altas velocidades.

Conclui-se então que é necessário uma forma de se ajustar automaticamente a velocidade de navegação, onde a necessidade de intervenção por parte do usuário seja a mínima possível.

1.2.2

Ajuste Correto dos Planos de Corte

Os planos de corte *near* e *far*, quando mal configurados também podem se tornar uma fonte de desorientação para o usuário. Esses planos, juntamente com os outros quatro planos que formam o *frustum de visão* (Figura 1.5), definem a região do espaço que pode ser vista. Qualquer objeto localizado fora desses limites não será renderizado pela placa gráfica. Dessa forma, a atribuição de valores incorretos pode fazer com que objetos não sejam exibidos na tela ou, ainda, que sejam parcialmente cortados, como mostra a Figura 1.6.

Os valores *near* e *far* também influenciam fortemente no uso eficiente do *buffer de profundidade*. Devido ao modo como o cálculo de perspectiva é feito, o *buffer de profundidade* possui uma precisão maior para objetos localizados perto do plano *near* do que para os localizados perto do plano *far*. De acordo com [21], sendo $r = \frac{\textit{near}}{\textit{far}}$, então $\log_2 r$ bits de precisão do *buffer de profundidade* são perdidos. Dessa forma, quanto maior for r , menor será a capacidade de se distinguir duas superfícies próximas entre si. O resultado

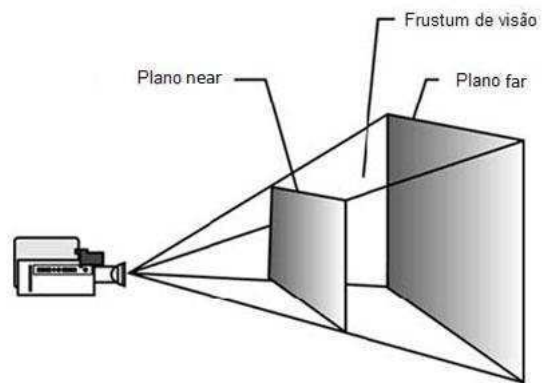
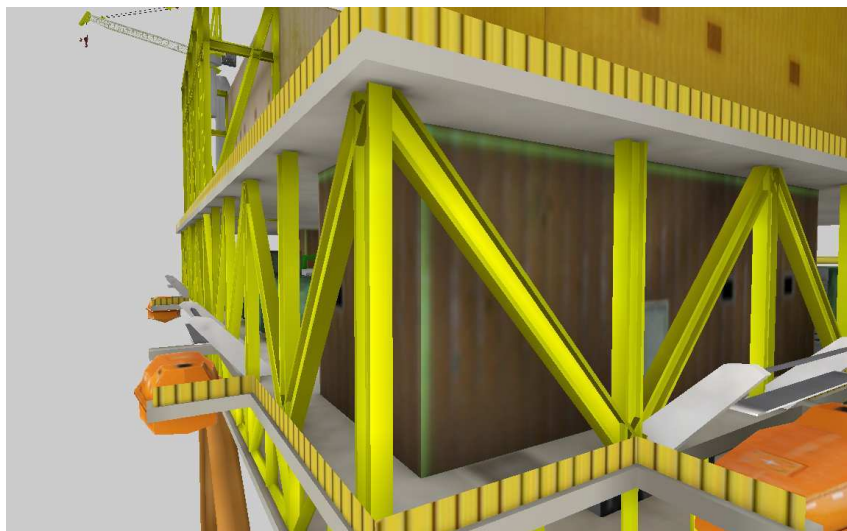


Figura 1.5: Frustum de visão



1.6(a): Plano *near* definido corretamente.



1.6(b): Plano *near* incorreto cortando do modelo.

Figura 1.6: Ajuste dos planos de corte.

disso é o aparecimento de artefatos (pontos que ficam piscando na tela) em objetos que estão mais distantes da câmera.

As soluções para esses dois problemas são conflitantes: por um lado, deseja-se aumentar o valor de r a fim de garantir que nenhum objeto seja cortado indevidamente. Por outro, o valor de r deve ser pequeno o suficiente para que não ocorram efeitos indesejados na visualização dos modelos. Esse dilema ocorre principalmente quando se deseja exibir cenas muito grandes, o que é justamente o caso do SiVIEP.

1.2.3

Tratamento de Colisão

Quando não há tratamento de colisão, torna-se possível atravessar objetos presentes na cena. Apesar de conveniente em alguns casos, essa situação geralmente configura um erro do ponto de vista do usuário.

Identificou-se que algumas pessoas, ao usarem o SiVIEP, sentiam desconforto ou ficavam desorientadas quando por algum motivo atravessavam paredes ou equipamentos presentes em uma plataforma. Outra situação em que esse efeito ocorria era ao navegarem entre duas camadas de um reservatório: os usuários não conseguiam se manter no espaço entre camadas e acabavam colidindo com essas.

Por fim, no futuro pretende-se que o SiVIEP tenha suporte a imersão com o uso de *caves*. Neste caso, colisões com o ambiente podem resultar em situações bem desconfortáveis [5], e o tratamento daquelas torna-se imprescindível. Ainda, atualmente o SiVIEP conta com suporte ao efeito de estereoscopia. Quando este está ativado, o ato de atravessar um objeto provoca uma sensação física nos olhos do usuário que pode ser bem desconfortável.

1.2.4

Senso de Orientação e Localização

Saber se orientar em um determinado ambiente é fundamental para o processo de navegação. Segundo conclui [7], sem indicações de direção, a desorientação inibe uma navegação eficiente e esse problema é pior em mundos virtuais maiores ou com pouca organização estrutural.

A desorientação também pode ser causada pelo uso incorreto das ferramentas de navegação. Um problema identificado no SiVIEP revelou que alguns usuários têm dificuldade em lidar com a ferramenta de centro de rotação. Essa permite escolher um novo centro de rotação que, por sua vez, é usado pela ferramenta *examinar*.

O problema é que a maioria das pessoas esqueciam de escolher um novo centro de rotação antes de iniciar o uso da ferramenta *examinar*. Como exemplo, considere inicialmente que o centro de rotação está localizado em um objeto $p1$. O usuário decide então navegar até outro objeto $p2$ usando a ferramenta *voar*. Ao chegar lá, tenta examinar $p2$ rotacionando a câmera em torno desse, mas é surpreendido por um movimento que considera estranho. Isso ocorre pois a rotação está sendo feita em torno de $p1$, não $p2$, uma vez que o centro de rotação não foi devidamente reconfigurado.

Concluindo, deve-se fornecer ao usuário alguma indicação que o permita se localizar em relação ao ambiente. É necessário também que haja uma forma de se garantir o estado correto do centro de rotação.

1.3 Objetivo

O objetivo deste trabalho é propor e implementar soluções para os problemas que foram abordados anteriormente. Tais soluções devem atender da melhor forma a dois requisitos principais:

1. Serem o mais automatizadas possíveis, de modo que exijam o mínimo de intervenção por parte do usuário.
2. Serem independentes dos tipos de modelos a serem visualizados. Ou seja, as soluções não devem se basear em características únicas de cada modelo. Isso permite que novos tipos de objetos sejam visualizados no futuro sem que haja necessidade de criar novas soluções.

Nesse sentido, foram estudados vários trabalhos de outros pesquisadores visando encontrar possíveis soluções. Dentre esses, o trabalho de [19] se revelou como a melhor das alternativas.

[19] abordam as questões do controle de velocidade de navegação em AMEs, do ajuste dos planos de corte e da detecção de colisão entre a câmera e o ambiente virtual. Para isso, eles criaram uma estrutura de dados chamada de *cubo de distâncias* (*Cube Map*, no original em inglês), que serve como base de todas as técnicas apresentadas por eles.

Neste trabalho foram implementadas algumas técnicas descritas por [19] e são propostas também outras soluções baseadas na estrutura *cubo de distâncias*.

1.4

Estrutura da Dissertação

Este trabalho está organizado em 5 capítulos. No próximo capítulo são discutidos outros trabalhos relacionados ao tema desta dissertação. No Capítulo 3 são apresentadas as soluções implementadas para resolver os problemas descritos nas seções anteriores. O Capítulo 4 apresenta resultados de desempenho e testes com usuários. Por fim, o Capítulo 5 conclui este trabalho e apresenta algumas propostas de trabalhos futuros.