HyperCAL^{3D}, uma Ferramenta Computacional para o Apoio do Processo de Ensino-Aprendizagem de Geometria Descritiva

F.G. Teixeira[a,b], **S.L. dos Santos**[b]

[a]***fabiogt@ufrgs.br***

[b] *Programa de Pós-Graduação em Design, Departamento de Design e Expressão Gráfica, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil*

Resumo

Este trabalho apresenta o HyperCAL^{3D}, um aplicativo de apoio ao ensino de Geometria Descritiva, através do estudo de objetos sólidos. São descritas a metodologia utilizada para a sua implementação e as principais funcionalidades do aplicativo. Foi realizada uma seleção de conceitos a fim de determinar a estrutura funcional que o software deveria atender. A partir disso, foram modeladas as principais funcionalidades através de processos de Geometria vetorial equivalentes aos utilizados na Geometria Descritiva. Dentre as principais funcionalidades apresentadas destacam-se: o processo de projeção, a representação de linhas ocultas no modelo tridimensional e nas projeções, vistas auxiliares sucessivas em tempo real e em 3D, representação em épura e o processo de interseção. Todas estas ferramentas são implementadas em um aplicativo que auxilia o processo de aprendizagem dos alunos e os procedimentos didáticos dos professores.

Palavras-chave: HyperCAL^{3D}, Geometria Descritiva, Aprendizagem Assistida por Computador.

HyperCAL^{3D}, A Computational Tool to Support the Teaching and Learning Process of Descriptive Geometry

Abstract

This work presents the HyperCAL3D, an application to support the teaching and learning process of Descriptive Geometry through the study of solid objects. The methodology used for its implementation and the main features of the application are described. A selection of concepts was carried out to determine the functional structure of the software should have. From this, the main features were modeled through processes of vector geometry equivalent to that used in Descriptive Geometry. The main features introduced include: the projection process, the representation of hidden lines in three-dimensional model and the projections, successive auxiliary views in real time, and the intersection process. All these tools are implemented in an application that aids the learning process of the students and the teaching procedures of professors.

Keywords: HyperCAL^{3D}, Descriptive Geometry, Computer-based Learning.

1. INTRODUÇÃO

A Geometria Descritiva (GD) é uma disciplina fundamental para a atividade de projeto. Sistematizada por Gaspard Monge no final do século XVIII, a GD é base das vistas ortográficas e do Desenho Técnico (DT) moderno (ROVIDA, 2013, p.72). O desenvolvimento da GD e, consequentemente, do DT alavancou a revolução industrial, pois possibilitou o desenvolvimento de máquinas e equipamentos com mecanismos de precisão, o que era impossível até então, sem métodos gráficos para o projeto geométrico de peças e estruturas. Por ser tão importante para a atividade de projeto, a GD é parte dos currículos dos cursos de Engenharia e Arquitetura desde, praticamente, a sua criação e, mais recentemente, dos cursos de Design.

Apesar de sua importância, os currículos destes cursos vêm tendo carga horária de GD reduzida e, em muitos casos, até eliminada. Com o advento dos sistemas CAD (*Computer-Aided Design*), há quem considere que não há mais necessidade de aprender a teoria dos processos gráficos, pois estes são feitos de forma automática pelos programas computacionais. No entanto, trata-se de uma ilusão, pois os sistemas CAD não fazem os projetos pelo projetista. É este que deve possuir o raciocínio espacial para projetar em 3D. O CAD é uma valiosa ferramenta, mas sem um usuário capacitado, com conhecimento dos processos gráficos e raciocínio espacial apurado, torna-se inútil. Portanto, eliminar a Geometria Descritiva dos cursos, acarretará na formação inadequada de projetistas, pois estes não terão domínio sobre o espaço, o que deveria ser uma das habilidades primordiais de profissionais que projetam (Engenheiros, Arquitetos e Designers).

A carga horária reduzida também é um problema para o ensino de GD, pois trata-se de uma disciplina de conteúdo denso e que exige grande abstração do aluno. Assim, a carga horária deveria ser suficiente para um treinamento adequado de todos os conceitos e processos da GD. Nos cursos da Universidade Federal do Rio Grande do Sul (UFRGS), em Porto Alegre, a carga horária é, em geral, de 60 horas. Porém, alguns cursos dispõem somente de 30 horas. Diante desta baixa carga horária, não há como esperar um nível de

aprendizado elevado, pois a abordagem tradicional da Geometria Descritiva (GD) exige um alto grau de abstração desde o início e falta tempo para aplicações contextualizadas dos conceitos.

Apesar de ter quase 220 anos, pouco se inovou no ensino de GD ao longo destes mais de 2 séculos. Há iniciativas isoladas, porém não houve ainda uma mudança de paradigma que revolucionasse o ensino da GD. As principais iniciativas são baseadas no uso de tecnologias computacionais através de modelos virtuais bi e tridimensionais que ilustram os processos da GD tradicional (RODRIGUES, et al., 2007; TEIXEIRA, et al., 1999; SEABRA & SANTOS, 2013).

Em contraponto, alguns autores propõem formas alternativas de uso da tecnologia, fugindo das meras ilustrações através da busca por novas formas de ensinar a GD. Um exemplo é o programa *Descriptive Geometry* (PLAVJANIK, 1999), o qual utiliza um ambiente onde são resolvidos problemas de GD através de uma linguagem de programação dedicada, baseada na linguagem C. Jr. (1998) e Chen (2000) apresentam iniciativas de uso de sistemas CAD para descrever processos complexos na GD.

No entanto, é partir do trabalho de Teixeira et al.(2006) que surge a promessa de uma mudança de paradigma no ensino da GD. Neste trabalho, os autores propõem uma nova abordagem (totalmente baseada em objetos sólidos), uma nova nomenclatura (alinhada aos conceitos da geometria tridimensional) e um processo que busca aumentar a experiência do aluno com os elementos e processos tridimensionais de forma integral. A abordagem proposta busca a experiência concreta antes da introdução de abstrações.

Com a proposta da nova abordagem para o ensino de GD, havia um problema: como proporcionar uma experiência concreta e, ao mesmo tempo, ensinar os principais conceitos da Geometria Descritiva? A hipótese para a solução foi que a experiência concreta e o ensino deveriam se dar das mais variadas formas, como modelos físicos, modelos virtuais e simulações computacionais dos processos gráficos em um ambiente tridimensional interativo.

Assim, o objetivo principal deste trabalho é o desenvolvimento de um programa computacional interativo que proporcione uma experiência concreta e também proporcione o entendimento dos conceitos fundamentais da GD.

O resultado é o HyperCAL^{3D}, um programa de apoio à aprendizagem de GD. Este programa computacional é inovador pois foi desenvolvido especificamente para o ensino de GD e simula os seus processos em um ambiente tridimensional em tempo real através da manipulação direta dos elementos. Além disso, todo o programa, que é uma aplicação direta de Computação Gráfica, foi desenvolvido utilizando os conceitos da Geometria Descritiva em uma abordagem vetorial. Portanto, o projeto do programa é totalmente baseado nos conceitos de GD.

O HyperCAL^{3D} foi concebido para ser utilizado tanto pelos professores, para a criação de exercícios e exposição de conteúdo, como pelos alunos, para a solução de problemas e a visualização dos processos de forma interativa. Este artigo apresenta os principais conceitos utilizados no desenvolvimento do HyperCAL^{3D}, além de mostrar exemplos de aplicações do programa para o ensino de GD.

2. METODOLOGIA

A metodologia para o desenvolvimento do HyperCAL^{3D} envolveu uma análise vetorial dos processos gráficos da GD a fim de sintetizar a simulação computacional dos mesmos. A seguir são descritas as principais etapas do processo de desenvolvimento:

- Seleção dos conceitos de GD a serem suportados pelo programa;
- Análise dos processos gráficos da GD através da decomposição em suas funções primárias;
- Síntese das operações vetoriais equivalentes às funções básicas dos processos gráficos;
- Desenvolvimento de uma estrutura de dados para permitir a representação dos objetos gráficos e as operações vetoriais associadas;
- Implementação computacional através de uma interface gráfica interativa.

A seguir serão detalhados alguns dos principais aspectos destas etapas metodológicas.

2.1 Seleção dos Conceitos

A etapa de seleção de conceitos está relacionada com o currículo da disciplina Geometria Descritiva que inclui os conceitos tradicionais da GD e mais alguns conceitos normalmente não relacionados a esta disciplina (Quadro 1).

Quadro 1: Lista de conceitos escolhidos para o HyperCAL^{3D}.

Conteúdo	Tópicos
Sistema Mongeano	Sistemas Projetivos; Sistema Mongeano [Dupla Projeção, Sistema de Referência]: características, nomenclatura, coordenadas, planificação do Sistema (épura).
Conceitos Básicos	Tipos de projeções de faces e arestas: projeções acumuladas, reduzidas e em verdadeira grandeza; Pertinência envolvendo retas (arestas); Posições entre retas (concorrentes, paralelas e reversas. Representação de planos (faces): Tipos de projeções de planos; Pertinência envolvendo planos (faces) Posições particulares de arestas e faces em relação ao Sistema de Projeção Mongeano: Os sete tipos de faces e sete tipos de arestas. Visibilidade de arestas e faces.
Mudanças de Sistema de Referência Primárias (MSRP)	Princípios da MSRP; Aplicações da MSRP para: obter VG de arestas oblíquas, obter VG de faces acumuladas de sólidos. Aplicações da MSRP para acumular arestas e acumular faces.
Mudanças de Sistema de Referência Sucessivas (MSRS)	Princípios das MSRS; Aplicações de MSRS para: obter VG das faces de um sólido; obter ângulo entre arestas; obter ângulo entre faces. Aplicações de MSRS para: Vista em perspectiva Axonométrica. Exercícios. Aplicações de MSRS para: obter vistas principais de um sólido.
Modelagem avançada de Sólidos	Perpendicularismo e Paralelismo de retas e planos no contexto dos sólidos. Distância. Extrusão. Interseções e Cortes. Interseção reta/plano; Interseção plano/plano; Cortes de sólidos.
Método da Revolução	Método da Revolução. Aplicações no contexto de objetos sólidos

Fonte: o autor.

Todos os conceitos básicos tradicionais foram contemplados, incluindo projeção, sistemas projetivos, sistema de projeção Mongeano e sua planificação em épura. Os conceitos tradicionalmente conhecidos como mudanças de plano são chamados aqui de Mudanças de Sistema de Referência (MSR). Esta mudança de nomenclatura serve para

aproximar a GD da matemática e da computação gráfica que usam as mesmas expressões.

Outros conceitos adicionados são as perspectivas como uma aplicação das MSR. Esta abordagem permite relacionar os conceitos de GD com os conceitos da perspectiva, demonstrando a origem deste conhecimento. A perspectiva geométrica com base nos princípios óticos foi sistematizada por Filipo Bruneleschi, no século XV, o qual fundou as bases para outros matemáticos, como Monge, desenvolverem seus trabalhos (Walker, 2003, p.62). Aqui estão incluídas as perspectivas axonométricas e as cônicas. Além disso, o desenho em perspectiva permite ver em 3D, o que faz parte da mudança de paradigma proposta.

Ou tópico relacionado à perspectiva é a obtenção das vistas principais de um sólido através de Mudanças de Sistema de Referência Sucessivas (MSRS). Neste caso, parte-se da perpectiva para obter as vistas ortográficas principais.

Os demais conceitos relacionados no Quadro 1 não apresentam novidades importantes em relação aos conceitos tradicionais da GD.

2.2 Análise dos Conceitos

Através de uma cuidadosa análise dos conceitos, foi observado que o conceito fundamental é a projeção, pois é a base para a representação e para as transformações. Em seguida, os conceitos de revolução e épura que estão diretamente associados à projeção. Em um outro nível está o conceito operacional de Mudança de Sistema de Referência (MSR), o qual utiliza o conceito de projeção para geral as transformações. Em um último nível estão os conceitos de interseção, cortes e seções, os quais utilizam o conceito da MSR como base. A Figura 1 apresenta um diagrama das relações analisadas.

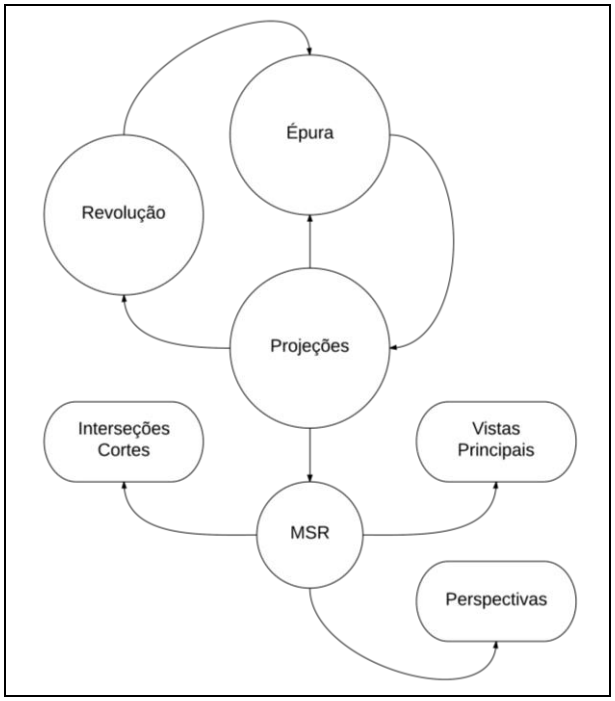


Figura 1: Relações entre os conceitos de GD proposta. Fonte: o autor.

Na próxima etapa, estes conceitos são desdobrados em processos vetoriais com o objetivo simular em um ambiente computacional os processos gráficos da GD.

2.3 Síntese das Principais Operações Vetoriais

As principais operações vetoriais construídas a partir da análise têm como objetivo obter: projeções; Mudanças de Sistema de Referência; revolução de objetos em torno de um eixo; representação em Épura; visibilidade de arestas e faces; interseções e cortes.

2.3.1 Sistema de Projeção Virtual

O processo utilizado no HyperCAL^{3D} para realizar as projeções utiliza a geometria vetorial para realizar o mesmo processo gráfico utilizado na GD. O algoritmo desenvolvido para realizar a projeção considera que cada plano de projeção possui um Sistema de Referência Local (SRL) associado, onde o plano XY do SRL é coincidente com o plano de projeção. Assim, cada vértice do objeto a ser projetado tem suas coordenadas computadas no SRL do plano de projeção. A projeção de cada vértice é computada considerando como zero a componente de Z das suas coordenadas no SRL e, em seguida, transformando novamente para o Sistema de Referência Global (SRG). A projeção do objeto é gerada unindo as projeções dos vértices segundo a topologia do modelo sólido. Este procedimento é extremamente simples e eficiente para a geração das projeções.

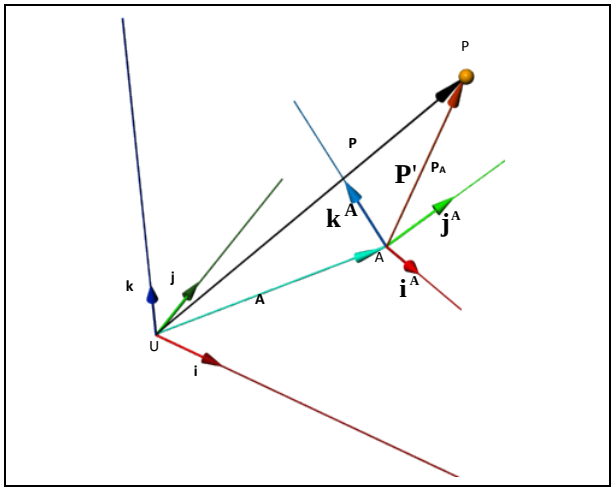


Figura 2: Esquema vetorial para o cálculo das coordenadas no SRL. Fonte: o autor.

A Figura 2 ilustra o processo vetorial de transformação de coordenadas de um SRG para um SRL. O processo para calcular as coordenadas no SRL é fundamentado no produto escalar entre o vetor posição do vértice em relação à origem do SRL e os vetores unitários dos eixos locais XL, YL e ZL. Cada um destes produtos escalares corresponde às coordenadas locais no eixo correspondente.

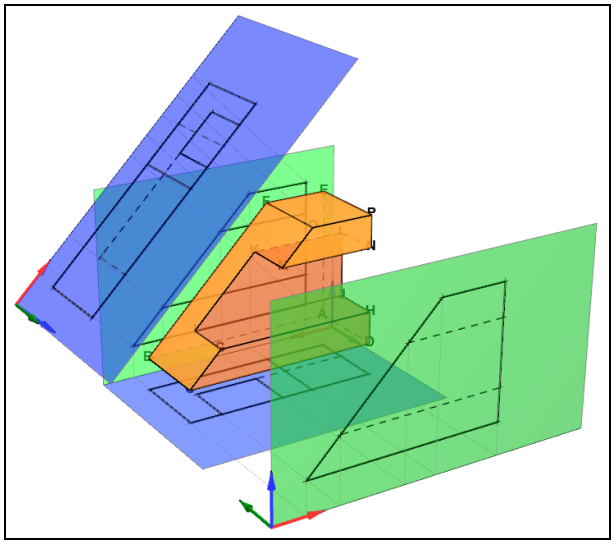


Figura 3: Exemplos de pares de planos de projeção e seus sistemas de referência. Fonte: o autor.

O sistema de projeção desenvolvido para o HyperCAL[3D] utiliza a mesma lógica do sistema Mongeano, onde cada plano de projeção é ortogonal aos planos adjacentes, que forma um diedro com dupla projeção. O sistema desenvolvido permite um número ilimitado de planos de projeção, de tal forma que os Sistemas de Referência (SR) sejam sempre definidos por pares de planos, um horizontal (PH) e outro frontal (PF). Cada plano de projeção criado dispõe de uma estrutura de classes e métodos que reproduzem as operações vetoriais descritas para a realização das projeções.

A Figura 3 ilustra muito bem esta organização, onde é possível observar os dois planos de projeção originais, com os eixos do SR e dois outros planos, cada um com o SR correspondente, os quais são compostos por pares de PH e PF, representados em azul e verde, respectivamente.

2.3.2 Mudança de Sistema de Referência

A Mudança de Sistema de Referência (MSR), que é conhecida tradicionalmente como Mudança de Plano de Projeção, utiliza o mesmo processo de projeção descrito no item anterior. Neste caso, este procedimento é o equivalente vetorial ao processo gráfico de traçar linhas de chamada perpendiculares à linha de terra e transportar as coordenadas do plano de projeção anterior. Assim, o Sistema de Projeção desenvolvido para o HyperCAL[3D] já comporta nativamente o processo de MSR, pois admite múltiplos planos de projeção simultâneos. A grande inovação aqui está na interface, a qual permite que o usuário insira um novo plano de projeção de forma direta e manipule a sua posição em tempo real de forma interativa com o *mouse*.

O posicionamento de um novo plano de projeção, que também é baseado em processos vetoriais, utiliza as seguintes diretrizes:

- O novo plano é sempre perpendicular a um plano de projeção existente;
- O vetor normal do novo plano de projeção é sempre paralelo ao plano de referência e sua direção é definida pela projeção do centro do sólido sobre o plano de referência;
- O plano de projeção é sempre orientado de tal forma que o sólido sempre esteja posicionado no primeiro diedro.

Estes parâmetros permitem posicionar o novo plano de projeção de forma interativa e em tempo real, pois os cálculos vetoriais envolvidos são simples e possibilitam, ainda, a visualização dinâmica das projeções resultantes. Assim, o posicionamento de um novo plano de projeção é definido pelo seu vetor normal e pela origem do SRL.

O Quadro 2 descreve o processo interativo de definição de um novo plano de projeção e como o programa processa as informações fornecidas pelo usuário para definir o posicionamento e a orientação do novo plano de projeção utilizando procedimentos de geometria vetorial.

A direção do plano é definida pela posição selecionada pelo usuário. No entanto, há uma ferramenta que permite definir determinadas direções com grande precisão, de forma a atender às situações específicas da GD. Para isso, o usuário seleciona uma determinada projeção no plano de referência, a qual irá orientar o plano de projeção de forma que este fique paralelo ou perpendicular à mesma. O processo utiliza um algoritmo que calcula o ângulo entre o vetor normal do plano e a direção selecionada através do produto escalar dos vetores. O plano de projeção trava em ângulos próximos à 90º ou Oº, com uma determinada tolerância.

Quadro 2: Processo interativo para definir um plano de projeção.

- O usuário seleciona o plano de projeção de referência:
  - O vetor normal do plano de referência define o eixo Y_L do novo plano de projeção.
- O usuário define a posição do centro do novo plano de projeção sobre o plano de referência:
  - É criado um vetor, definido pela direção entre a projeção do centro do sólido no plano de referência e a posição definida pelo usuário, o qual irá corresponder ao eixo Z_L.
  - O eixo X_L é definido pelo produto vetorial $Z_L = Z_L \times Y_L$.
  - O SRL é deslocado de tal forma que fique sempre evidente um triedro direto e um sistema de primeiro diedro.

A Figura 4a ilustra o processo de posicionamento do plano auxiliar de forma que o mesmo fique perpendicular a uma direção selecionada, destacada com círculo na figura. Neste caso, o objetivo é acumular uma das arestas do sólido e, consequentemente, a face que a contêm. Na Figura 4b, é acrescentado um segundo plano auxiliar, que é perpendicular ao primeiro plano auxiliar, e paralelo à face acumulada na MSR primária, a qual é selecionada para definir a direção paralela do plano.

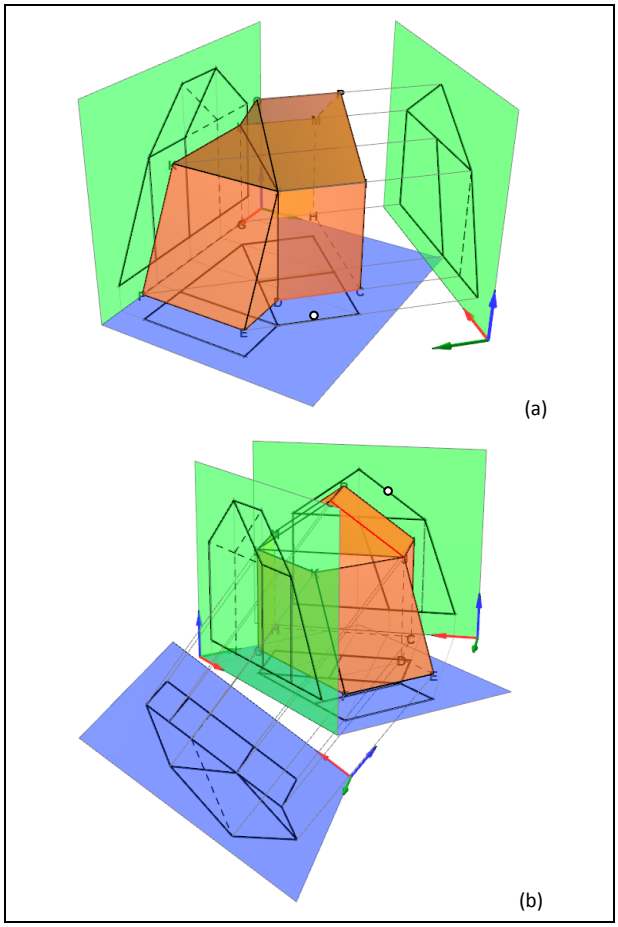


Figura 4: a) Posicionamento do plano auxiliar de projeção perpendicular a uma direção selecionada. b) Posicionamento de um segundo plano auxiliar perpendicular ao anterior e paralelo à projeção da face acumulada. Fonte: o autor.

2.3.3 Representação de Linhas Ocultas

A correta representação de objetos sólidos, implica na representação adequada de linha ocultas, aquelas que não são visíveis pelo observador. No HyperCAL[3D], a visibilidade é analisada no modelo 3D e nas suas projeções. A solução adotada utiliza abordagens semelhantes nas duas situações e,

o mais importante, utiliza a mesma técnica da GD para verificar a visibilidade de linhas em épura.

Para verificar a visibilidade de uma linha em um plano de projeção na GD, são analisados os cruzamentos de linhas e, nos pontos de interseção das projeções, é verificada qual linha está mais distante do plano de projeção (ou mais próxima do observador). Esta é a linha visível e a outra, consequentemente, é invisível. Este processo é amplamente utilizado no ensino de GD e na solução de problemas de visibilidade nas projeções de objetos. No HyperCAL^{3D}, foi adotada uma técnica semelhante.

A visibilidade do modelo sólido utiliza ferramentas da biblioteca OpenGL para implementar um algoritmo do tipo *ray casting* (WATT, 1993, p.371), que traça uma reta a partir de qualquer objeto até à câmera e verifica a interseção desta com os objetos da cena virtual. Assim, é possível verificar se existem faces ou arestas entre o ponto de origem da reta e a câmera e as distâncias entre eles. Se houver, significa que o ponto não é visível. Caso contrário, o ponto é visível. Assim, para cada aresta do modelo, é feita uma amostragem de pontos e verificada a sua visibilidade. Podem haver três situações de visibilidade: totalmente visível, totalmente invisível e parcialmente visível. Nos dois primeiros casos, o processamento é concluído com verificação dos pontos amostrais e a visibilidade da aresta define o tipo de linha a ser utilizado na representação, linha contínua, para visível, e tracejada, para invisível.

No caso de a aresta ser parcialmente visível, é necessário determinar o ponto de mudança de visibilidade (PMV), para definir corretamente a porção com linha tracejada e aquela com linha cheia. O processo utilizado para encontrar o PMV da aresta é uma bissecção da aresta, repetindo de forma iterativa o processo descrito anteriormente para a verificação da visibilidade da aresta. O trecho entre os dois pontos cujas visibilidades são distintas é dividido ao meio e são verificadas as visibilidades das extremidades dos dois trechos resultantes. O trecho que possui extremidades com visibilidades diferentes é o escolhido para reiniciar o processo de bissecção. Este processo é repetido até que o trecho atinja uma dimensão inferior ao limite de precisão determinado.

A implementação deste algoritmo se mostrou apropriada à aplicação interativa e com resultados muito precisos. A Figura 5 mostra a representação das linhas ocultas de um modelo em diferentes pontos de vista. É possível observar uma linha destacada onde a visibilidade é alterada de acordo com a posição da câmera. Na Figura 5a, há uma aresta marcada totalmente visível e outra parcialmente visível. Na Figura 5b, em um outro ponto de vista, a aresta que era visível fica parcialmente visível, e a aresta que era parcialmente visível torna-se invisível. Parca cada vista, é necessário que todas as informações de visibilidade sejam reprocessadas.

A visibilidade do modelo pode ser atualizada em tempo real durante a movimentação da câmera pelo usuário. No entanto, por padrão, a visibilidade é desligada durante o movimento da câmera e novamente ligada quando a câmera para. Isto é feito para evitar uma queda no desempenho em computadores mais lentos, principalmente na visualização de modelos complexos, com muitos vértices e muitas faces.

Nas projeções, a visibilidade é realizada de forma similar à do modelo tridimensional e utiliza este como parâmetro de referência. Os pontos de mudança de visibilidade das arestas são determinados de forma exata a partir das interseções entre as projeções das arestas. Os pontos de interseção subdividem as projeções das arestas em trechos que podem ser visíveis ou invisíveis.

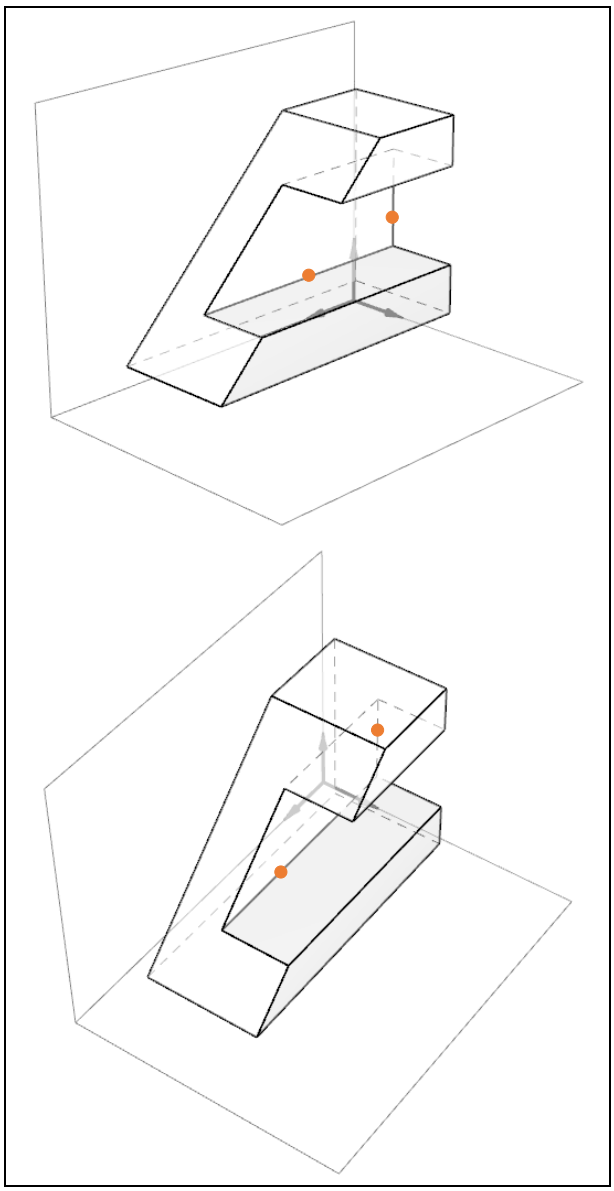


Figura 5: Exemplo de arestas com visibilidade total, parcial e totalmente invisível. Fonte: o autor.

As visibilidades de cada trecho das projeções de todas arestas são calculadas utilizando a mesma abordagem baseada no *ray casting*, adotada para a visibilidade modelo tridimensional. No entanto, a reta para verificar a visibilidade de pontos ou trechos específicos é traçada de forma ortogonal ao plano de projeção (PP), iniciando em pontos das arestas do modelo e se afastando do plano de projeção onde se quer determinar a visibilidade. O processo é realizado a partir da verificação da ocorrência de faces que são interceptadas pela reta de verificação que se afasta do PP. Se a reta intercepta alguma face após se afastar do PP partindo de um trecho de uma aresta, significa que este trecho da projeção é invisível neste PP. A Figura 6 ilustra este processo de determinação de visibilidade, com a representação esquemática da reta traçada para verificar se há faces que cobrem a aresta.

A visibilidade das projeções é atualizada em tempo real durante o posicionamento do plano de projeção quando o mesmo é inserido. No entanto, após o posicionamento do plano de projeção, a visibilidade não é atualizada a menos que o modelo seja alterado. O reposicionamento da câmera não requer uma atualização da visibilidade das projeções, uma vez que a visibilidade das projeções só depende da posição do objeto em relação ao plano de projeção.

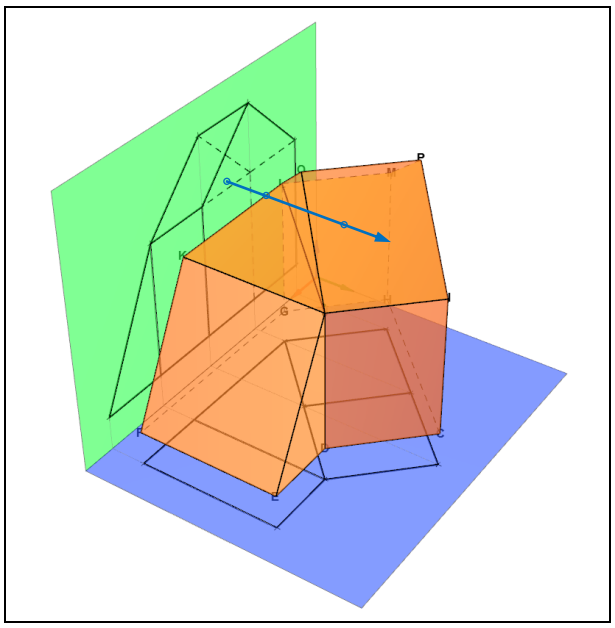


Figura 6: Determinação da visibilidade da projeção de uma aresta. Aresta invisível na projeção frontal. Fonte: o autor.

2.3.4 Representação em Épura

A representação automática em épura é uma ferramenta extremamente útil para o ensino de GD, pois permite a visualização do processo de representação convencional nas duas formas, em 3D e 2D. O HyperCAL^{3D} realiza este procedimento diretamente a partir do modelo tridimensional do sistema de projeção com todos os planos inseridos pelo usuário, simulando exatamente a situação da geração da épura na GD.

A representação em épura dos objetos é feita rebatendo os planos de projeção um a um, começando pelos últimos planos inseridos. Cada plano é rebatido sobre o seu plano antecessor. O rebatimento do plano de projeção consiste em um giro de 90º em torno da linha de terra, incluindo todas as projeções contidas no mesmo. Este processo de rebatimento consiste em um procedimento vetorial, onde é utilizada uma matriz de rotação (SCHNEIDER & EBERLY, 2003, p.857) acoplada à linha de terra (*eixo x*) de cada SR. Para que seja possível realizar todos os rebatimentos de forma robusta, os planos de projeção são organizados em uma estrutura de dados hierárquica do tipo árvore, onde a raiz é o plano horizontal. Todos os demais planos são ramos que convergem para a raiz.

Após montagem da árvore, cada plano é rebatido sobre o plano antecessor, na árvore, começando pelos ramos mais externos e terminando no plano horizontal, o único que não é rebatido. Para que um plano seja rebatido, toda a sua descendência já deve ter sido rebatida. Outra regra importante é que quando um plano é rebatido, todos os seus descendentes também o são. Esta sucessão de rebatimentos organizados e orientados por esta estrutura tipo árvore resultam na planificação completa do sistema de projeção. A Figura 7 apresenta um exemplo de épura obtida a partir de uma situação tridimensional onde foram realizadas duas MSR para a obtenção de VG de uma face oblíqua.

O rebatimento das projeções implica no rebatimento dos vértices, linhas de chamada e arestas. Os eixos dos SR associados a cada plano de projeção também são rebatidos. O resultado é uma representação em épura de grande qualidade gráfica e em alta resolução, com controle de *alias* e a possibilidade de *zoom* e *pan*. Na interface do HyperCAL^{3D}, a representação em épura é uma das páginas (*tabs*) da janela de visualização, as quais são ativadas a partir de abas no canto superior esquerdo da mesma. A qualquer momento, é possível alternar entre a visualização do modelo 3D e a representação em épura.

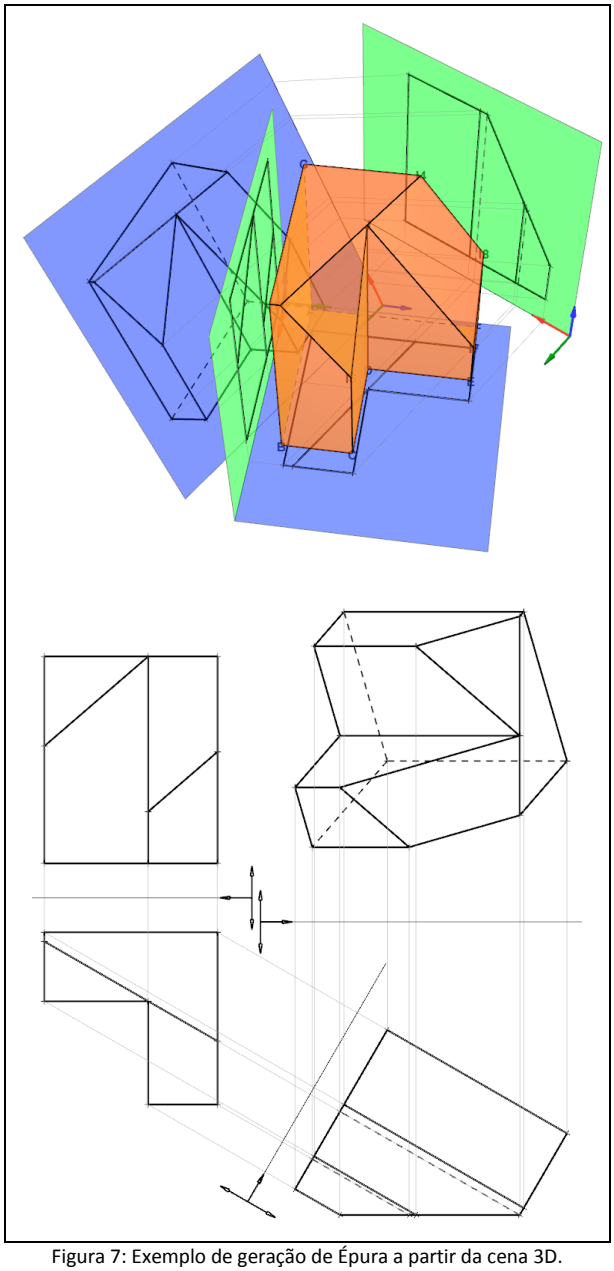


Figura 7: Exemplo de geração de Épura a partir da cena 3D. Fonte: o autor.

2.3.5 Cálculo de Interseção entre Planos

A capacidade de calcular interseções entre planos e entre reta e plano é fundamental para a GD. O conceito de interseção é fundamental em todos os processos da GD, incluindo a projeção, que é o fundamento mais elementar. Assim, o processo para determinar a interseção é o proposto em Teixeira (2003, p.38), onde um SR é acoplado ao plano secante de forma que o plano XY do SR fica coincidente com o plano.

A determinação da interseção é realizada computando as coordenadas dos vértices do sólido no SR auxiliar. Aquelas arestas que possuem vértices com coordenadas Z com sinais trocadas são cortadas pelo plano secante. O ponto exato de corte é determinado a partir da representação paramétrica das arestas. Com isso, é possível obter o ponto exato de interseção. A Figura 8 apresenta um esquema do processo.

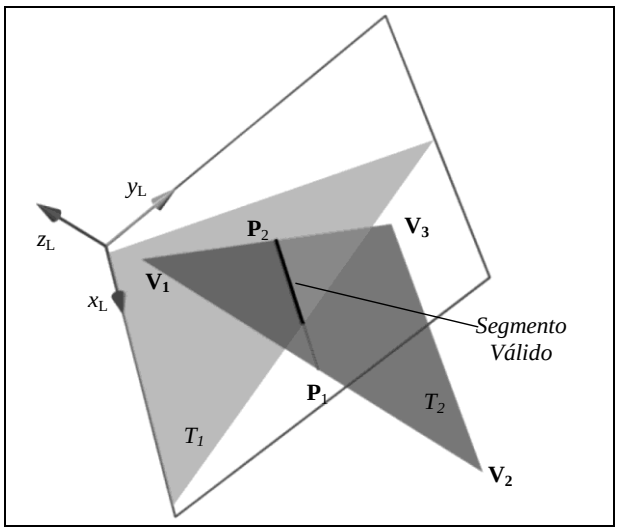


Figura 8: Esquema do processo de cálculo de interseção entre planos. Fonte: Teixeira (2003, p.38)

2.4 Estrutura de Dados

Todas as funcionalidades do HyperCAL^{3D} dependem de uma estrutura de dados que suporte tanto a representação dos objetos como as operações sobre os mesmos. A estrutura de dados desenvolvida foi definida como uma estrutura de classes hierárquica, que organiza todos os objetos e suas operações.

As classes geométricas básicas são Vértice, Aresta, Face e Sólido. Sendo que a classe Aresta possui dois Vértices como propriedades. A classe Face possui uma lista de Vértices e Arestas como propriedades. A classe Sólido possui uma lista de Vértices, uma lista de Faces e uma lista de Arestas como propriedades. A classe Sólido possui métodos que garantem a organização dos dados dos Vértices, Arestas e Faces. Além disso, esta classe possui métodos para a determinação da visibilidade no ambiente 3D.

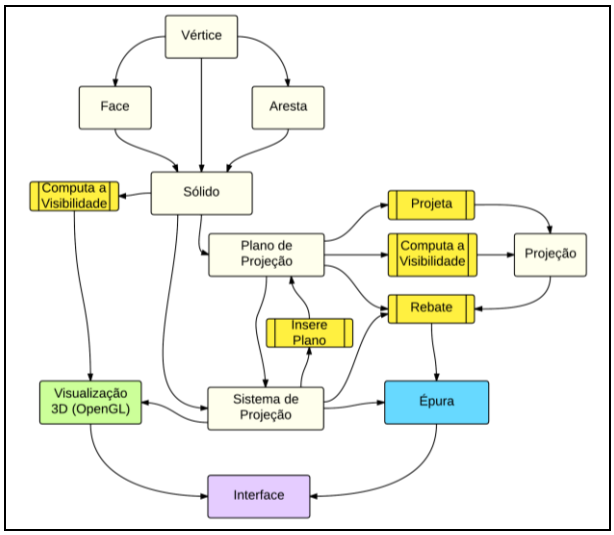


Figura 9: Estrutura de classes correspondente à estrutura de dados. Fonte: o autor.

O Sistema de Projeção consiste em uma classe que contém uma lista de objetos da classe Plano de Projeção. Esta classe, por sua vez, contém os métodos para projetar a geometria dos objetos geométricos, criando objetos da classe Projeção, os quais tornam-se propriedades do Plano de Projeção. Além disso, esta classe possui métodos que computam a visibilidade das linhas de projeção e para rebater o plano e todas as Projeções nele contidas. O Sistema de Projeção possui métodos para administrar os planos ele contidos, incluindo a inserção, a remoção e o rebatimento para gerar a Épura.

A Figura 9 ilustra a organização de classes do HyperCAL^{3D} de forma simplificada, mostrando apenas aquelas classes e métodos mais importantes para as principais características do programa.

A estrutura de classes completa possui dezenas de classes e centenas de métodos associados para dar conta de todas as funcionalidades projetadas para o Programa.

2.5 Ambiente Gráfico Tridimensional

A linguagem de programação escolhida para a implementação computacional é o Embarcadero Delphi®, que permite orientação a objetos e possui um desempenho compatível com aplicações interativas.

A tecnologia de processamento gráfico escolhida é a OpenGL® (Wright et al., 2007), que é uma biblioteca gráfica padrão na indústria da computação gráfica em seus diversos segmentos, incluindo PCs, estações gráficas e até dispositivos móveis. O OpenGL® disponibiliza uma API (*Application Programming Interface*) que pode ser utilizada com diversas linguagens de programação. No HyperCAL^{3D}, o OpenGL® é implementado por meio de uma biblioteca de alto nível, o GLScene, compatível com a linguagem Delphi, e que consiste em uma série de componentes, que utilizam a API do OpenGL®, e que podem ser utilizados para construir aplicativos gráficos executáveis.

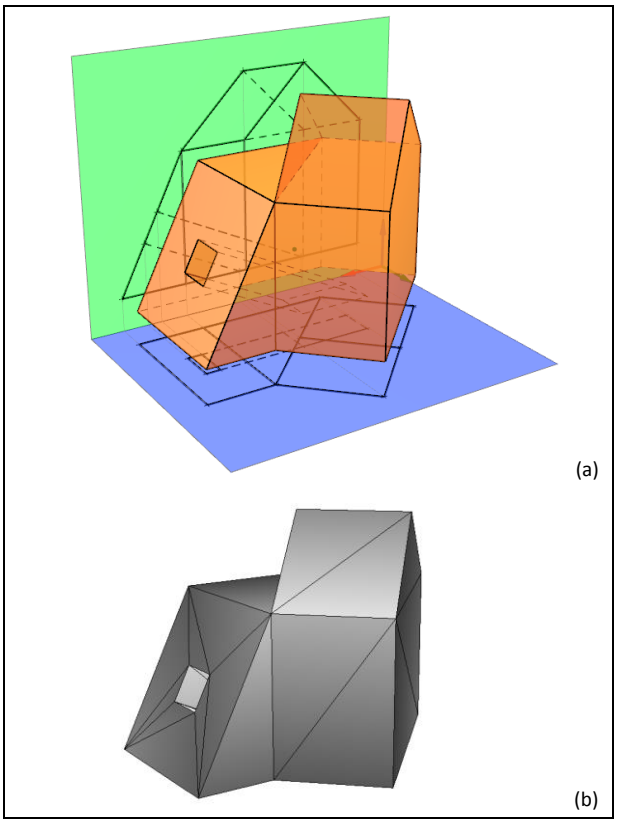


Figura 10: Representação dos sólidos no HyperCAL3D. a) Forma apresentada ao usuário. b) Faces representadas por malhas na estrutura de dados. Fonte: o autor.

A estrutura de dados para a representação dos modelos tridimensionais é baseada em uma representação hierárquica dos objetos. Assim, um sólido é dividido em faces e estas são subdivididas em malhas de triângulos. Isto é necessário, pois há características geométricas em sólidos não convexos, como os furos e reentrâncias, que podem inviabilizar a representação por um único polígono. Assim, cada face de

um sólido é, na verdade, uma malha de triângulos, o que permite a representação de faces com geometrias complexas.

A malha de triângulos é construída por um gerador de malha especialmente desenvolvido, baseado na técnica *advanced front* (Teixeira, 2003, p.9) e na triangulação de Delaunay (Agoston, 2005, p.722). A Figura 10 apresenta a representação de um sólido na forma que é apresentado ao usuário (Figura 10a) e a representação da malha de triângulos (Figura 10b), a qual não é percebida pelo usuário.

A representação dos planos de projeção não utiliza uma malha triangular, pois a forma é sempre retangular, podendo ser representado como um único polígono.

O ambiente tridimensional construído permite grande interatividade. É possível mover a câmera de forma dinâmica, selecionar objetos para os diversos fins no processo de interação com o usuário, como no caso da inserção de planos de projeção, por exemplo.

3. RESULTADOS E APLICAÇÕES

O resultado do processo de desenvolvimento é um programa com funcionalidades que permitem trabalhar com diversos conceitos da Geometria Descritiva (GD), possibilitando aos professores expor o conteúdo diretamente na sua interface. Várias são as aplicações de GD que podem utilizar o HyperCAL^{3D} como ferramenta de apoio:

- visualização de modelos 3D junto à épura, para entendimento da geometria;
- determinação de Verdadeira Grandeza (VG) de segmentos oblíquos por MSR primária;
- determinação de VG de face (plano) com projeção acumulada por uma MSR primária.
- determinação de projeção acumulada de plano oblíquo por MSR primária;
- determinação de VG de plano oblíquo por MSR dupla (sucessiva);
- determinação de distâncias entre ponto e reta, entre retas entre ponto e plano e entre planos;
- determinação de perspectiva axonométrica a partir de qualquer ponto de vista;
- determinação das vistas ortográficas principais de um sólido a partir de MSR sucessivas;
- determinação de interseção de entre sólidos e planos.

A seguir, são descritas a interface do Programa e algumas aplicações mais significativas do ensino da GD que podem ser utilizadas com o apoio do HyperCAL^{3D}.

3.1 Interface do HyperCAL^{3D}

A interface gráfica do HyperCAL^{3D} apresenta três áreas principais: área gráfica, painel de comandos e barra de ferramentas. A barra de ferramentas é a interface de acesso aos principais comandos.

O painel lateral é uma interface direta com diversas funcionalidades. Este painel contém várias páginas, acessadas pelas abas no seu topo. Aqui é possível inserir vértices (Pontos), conectividades das faces (Faces), posicionar a câmera de forma precisa (Câmera), configurar elementos da interface (Configuração), verificar características do modelo (Modelo) e determinar a interseção entre o sólido e um plano (Interseção).

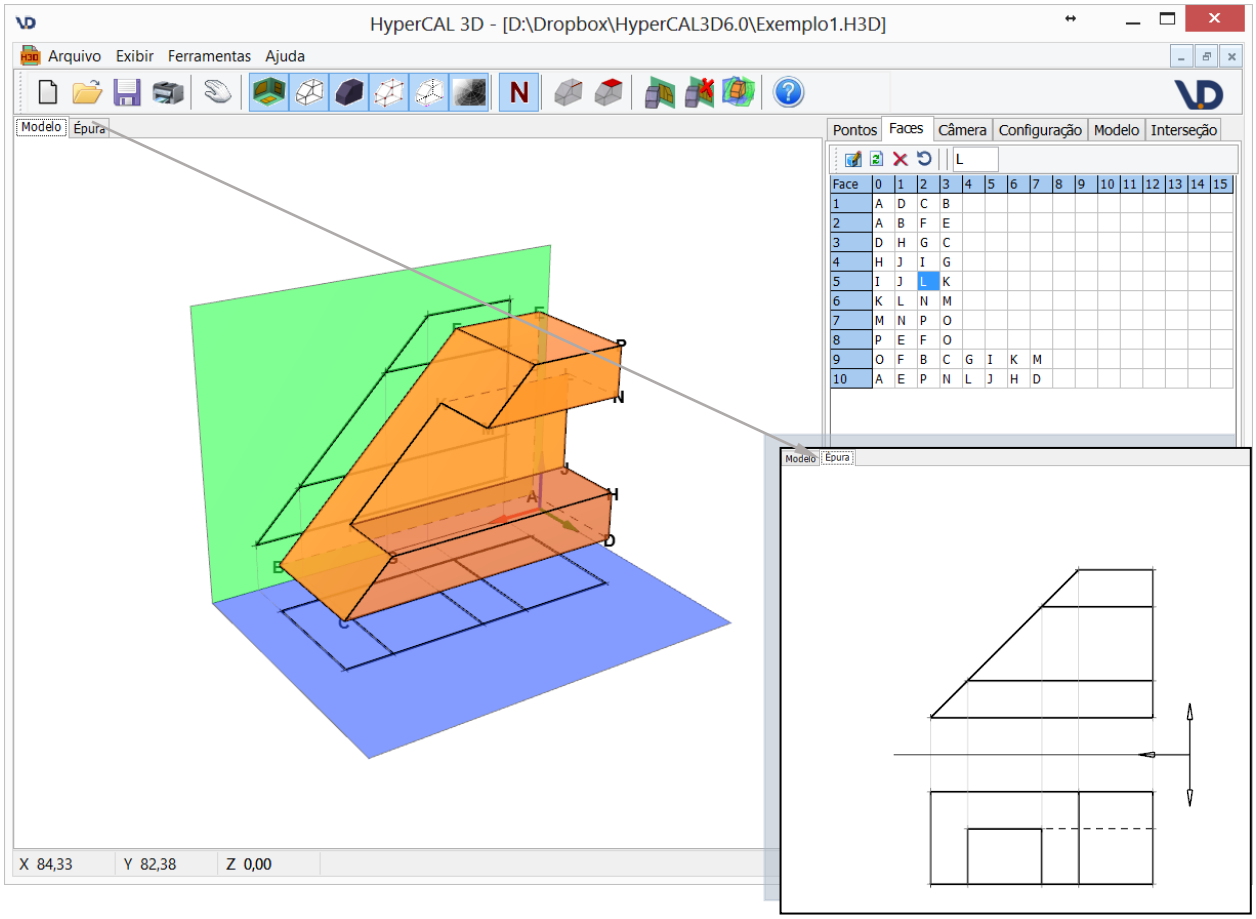


Figura 11: Interface do HyperCAL^{3D}, com o ambiente 3D e a épura (em destaque) , o painel lateral e e a barra de ferramentas e menu.
Fonte: o autor.

O processo de modelagem é totalmente manual. É necessário inserir coordenadas dos vértices na página Pontos e, em seguida, inserir as conectividades das faces na página Faces. As conectividades são a sequência de vértices que compõem uma face do sólido. O sólido será composto por um conjunto de faces. A orientação das conectividades define o vetor normal de cada face. Assim, por questões de coerência topológica, as faces devem possuir normais com a mesma orientação. Para alertar o usuário, quando a normal está na direção correta, a cor da face é laranja. Caso contrário, a cor fica amarela. Existe um comando, onde o usuário pode, facilmente inverter as conectividades e, consequentemente, a direção da norma de uma ou mais faces selecionadas, simultaneamente.

Na área gráfica, que é dominante na interface, há um painel com duas páginas, selecionáveis pelas abas no canto superior esquerdo). Na aba Modelo, é apresentada a interface 3D, que é a interface de trabalho principal. Nesta área, o ´solido é apresentado em 3D juntamente com os planos dos Sistema de Projeção. É aqui que são realizados os comandos interativos de posicionamento de câmera, seleção de objetos, inserção de planos de projeção para Mudanças de Sistema de Referência (MSR). Os objetos possuem cores específicas, conforme o seu tipo, mas é possível mudar a representação para preto branco. Na aba Épura, é acionada a página correspondente à épura da cena atual da página Modelo, onde os planos são rebatidos até que todas projeções fiquem no mesmo plano, exatamente como ocorre na GD. Tanto na área do Modelo quanto na Épura é possível comandos de visualização, como *Zoom* e *Pan* interativos, utilizando o rolete do mouse e o arrastar.com o botão central, respetivamente. É possível alternar entre o Modelo e a Épura de forma direta e rápida, permitindo a comparação entre as duas situações e proporcionando a reflexão do estudante.

Todo o processo de interatividade operacional se dá na área do Modelo. A única interatividade em épura se refere à visualização. Para o futuro, está planejado interatividade nos dois ambientes, 3D e 2D, de forma que será possível realizar operações em épura e verificar as consequências em 3D de forma que os comandos funcionem indistintamente nos dois ambientes de visualização, permitindo uma experiência mais completa dos processos da Geometria Descritiva.

3.2 Determinar VG de Segmentos Oblíquos.

O HyperCAL^{3D} não possui comandos específicos para uma determinada situação e não realiza automaticamente todas as tarefas para o usuário. Assim, a determinação da VG de um segmento de reta oblíquo é realizada através de uma MSR apropriada, a qual deve ser definida pelo usuário/estudante. Portanto, ainda é necessário compreender que o plano a ser inserida na MSR deve ser paralelo a uma das projeções do segmento de reta.

Da mesma forma, é decisão do usuário/estudante fazer uma MSR de forma a obter o ângulo em VG em relação ao Plano Horizontal (PH) de projeção ou ao Plano Frontal (PF) de projeção. Portanto, não se trata de o Programa resolver os exercícios para os estudantes. O Programa é uma ferramenta de apoio, mas sem o conhecimento de GD, não é possível resolver os problemas, com ou sem o Programa. A Figura 12 apresenta um exemplo de obtenção da VG de uma aresta oblíqua (GC) de um sólido. Neste caso, foi feita uma MSRP através de um novo PF paralelo à projeção horizontal da aresta. É possível ver o processo em 3D (Figura 12a) com o sólido completo, onde a VG da reta se projeta invisível no plano auxiliar. Em épura (Figura 12b), foi utilizada a opção de isolar a aresta, deixando apenas a aresta em estudo visível.

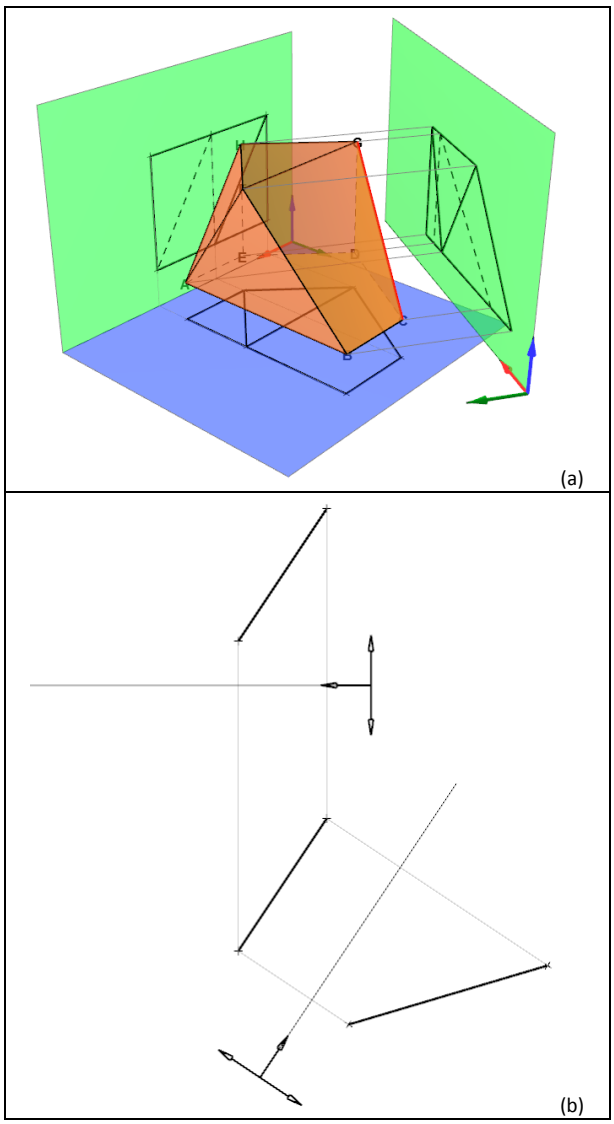


Figura 12: Exemplo de obtenção de VG de uma aresta oblíqua por uma MSRP em 3D (a) e em éura com a aresta isolada (b).
Fonte: o autor.

3.3 Determinar VG de Face com Projeção Acumulada

Determinar a VG de uma face de projeção acumulada (Hawk, 1962, p.29) é um problema que também pode ser resolvido com uma única MSR, desde que seja corretamente escolhida. Não existe um comando para isto no Programa. O usuário/estudante deve saber que um plano se projeta em VG quando sua projeção acumulada é paralela à Linha de Terra (LT – eixo X). E, ainda, o estudante deve compreender o que é e como identificar um plano de projeção acumulada.

Assim, para encontrar a VG de um plano de projeção acumulada utilizando o HyperCAL^{3D}, é necessário identificar a face e, então, realizar uma MSR acrescentando um plano perpendicular ao plano de projeção onde a face está acumulada. O posicionamento do plano deve ser paralelo à projeção acumulada do plano. Assim, o mesmo será projetado em VG neste plano auxiliar.

A Figura 13 ilustra este processo para um sólido com uma face de topo (perpendicular ao plano frontal de projeção). É possível observar, na vista em 3D (Figura 13a), que o plano auxiliar é realmente perpendicular ao PF e paralelo à projeção acumulada da face de topo. Em épura (Figura 13b) e com a face isolada, é possível observar que a nova LT (do novo SR definido pelo plano auxiliar) é paralela à projeção acumulada do plano de topo. Portanto, neste plano auxiliar, a face de topo se projeta em VG.

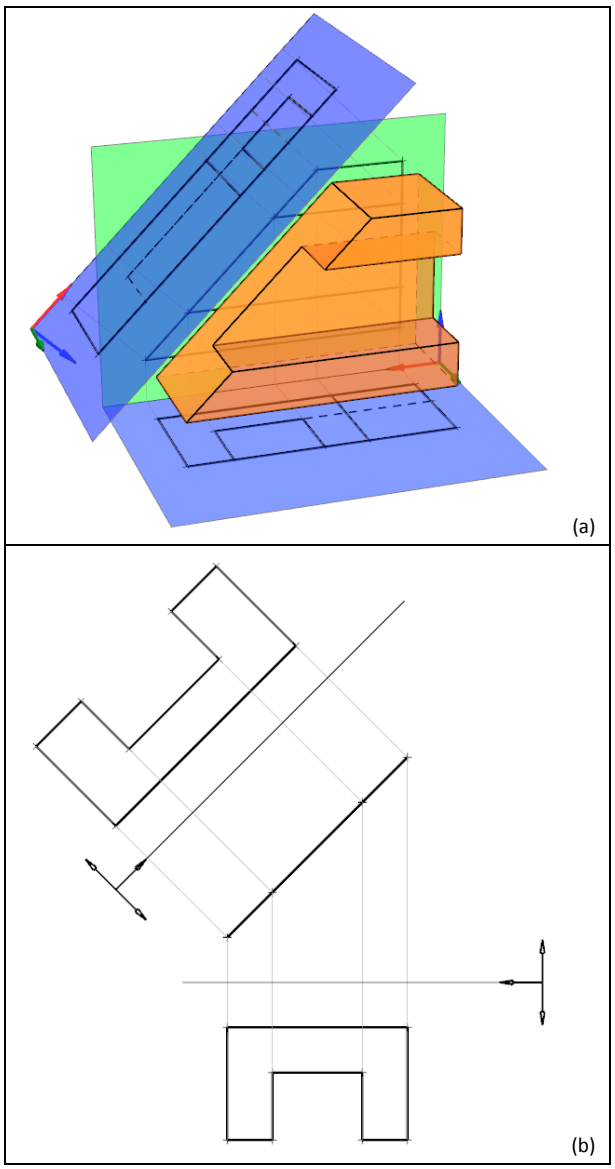


Figura 13: Determinação de VG de Plano de projeção acumulada em 3D (a) e em épura (b). Fonte: o autor.

3.4 Determinar Projeção Acumulada de Plano Oblíquo

A determinação de projeção acumulada de um plano oblíquo tem várias aplicações na GD. Seja para determinação de ângulo entre o plano e um plano de projeção, como um passo intermediário para a determinação de VG de plano oblíquo ou, ainda, em situações onde é necessário determinar distâncias em relação ao plano, a acumulação de um plano é um dos conceitos mais importantes para a solução de problemas envolvendo planos em GD.

Novamente, não há no HyperCAL^{3D} um comando específico para a determinação da projeção acumulada de um plano. É o usuário/estudante que deve fazer a MSR apropriada para atingir o resultado procurado. Assim, para acumular um plano, é necessário acumular uma reta do plano (Hawk, 1962, p.28), pois retas de projeção acumuladas só podem estar contidas em planos de projeção acumulada, uma vez que não é possível um plano que não seja perpendicular a um plano de projeção conter uma reta perpendicular a este mesmo plano de projeção.

Uma reta de projeção acumulada possui a projeção oposta em VG e perpendicular à LT. Assim, se uma reta está em VG, basta inserir um plano perpendicular à projeção em VG que a reta se projeta acumulada e qualquer plano que a contenha também é projetado acumulado. Assim, para acumular um plano, é necessário identificar ou representar uma reta em VG sobre o mesmo e, então realizar o processo descrito. A Figura 14 mostra um exemplo de obtenção de projeção acumulada de um plano oblíquo. No caso, uma face triangular, onde foi traçada uma reta horizontal contida no plano. Em seguida, foi feita uma MSRP inserindo um plano de projeção perpendicular à VG da reta horizontal. Com isto, a reta e o plano projetam-se acumulados no plano auxiliar.

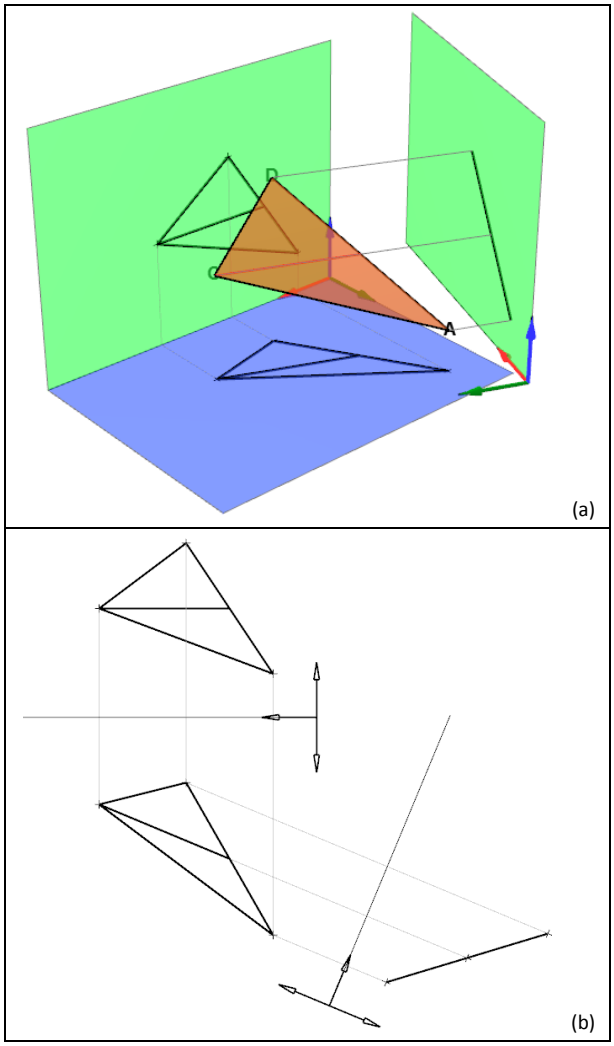


Figura 14: Determinação de projeção acumulada de plano oblíquo em 3D (a) e em épura (b). Fonte: o autor.

3.5 Determinar VG de Plano Oblíquo

Determinar a VG de um plano ou face oblíqua consiste em um dos principais problemas da GD. Pois, a partir de projeções reduzidas da face, é possível obter as suas reais dimensões apenas com procedimentos gráficos. Quando m plano se projeta em VG significa que o mesmo é paralelo ao plano de projeção. Isto fica indicado pela projeção adjacente ser acumulada e paralela à LT. Assim, para encontrar a projeção em VG de um plano oblíquo são necessárias duas MSR sucessivas. A primeira para acumular o plano e a segunda, utilizando uma LT paralela à projeção acumulada, para encontrar a VG.

No HyperCAL^{3D}, a segunda MSR é feita diretamente selecionando o como plano de base o plano auxiliar da MSRP e selecionando como direção a projeção acumulada do plano oblíquo obtida pela MSRP. A Figura 15a mostra o mesmo exemplo da Figura 14, onde o plano foi acumulado por uma MSRP, e foi acrescentado um novo plano paralelo à projeção acumulada do plano, no qual o mesmo se projeta em VG. É possível observar na épura (Figura 15b) a primeira LT, utilizada para acumular o plano e a segunda paralela à acumulação resultando na projeção em VG do plano. Este exemplo, muito

simples, com apenas uma face, foi apresentado para fins de compreensão do processo. No entanto, é possível trabalhar com sólidos complexos com a mesma agilidade.

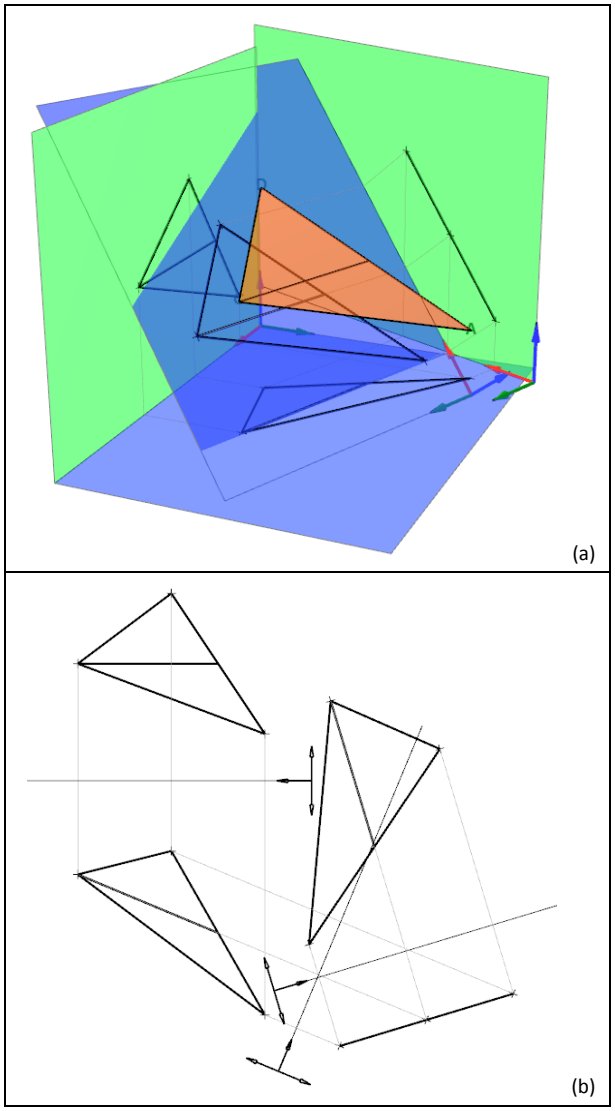


Figura 15: Determinação de projeção em VG de plano oblíquo em 3D (a) e em épura (b). Fonte: o autor.

3.5.1 Determinação de Vistas em Perspectiva

Perspectivas Axonométricas são aquelas obtidas a partir de projeções cilíndricas ortogonais, de forma que os três eixos principais do objeto se projetam não acumulados no plano de projeção. Como o sistema de projeção cilíndrico ortogonal é a base para o sistema Mongeano de representação, é possível obter uma vista em perspectiva axonométrica em projeção Mongeana escolhendo-se uma direção de observação apropriada. Esta direção de observação pode ser descrita como um vetor de observação (VO), pois a orientação também é importante, definindo de onde e para onde se observa. Se o objeto observado tem as direções principais paralelas aos eixos coordenados, o vetor de observação deve ser oblíquo para que a vista resultante seja uma perspectiva. A projeção em perspectiva é gerada quando o VO se projeta acumulado, indicando que a projeção resultante dada pela direção de observação do VO. Assim, dada uma direção de observação, é necessário realizar duas MSRS de forma a acumular o VO, resultando na vista em perspectiva.

No HyperCAL^{3D}, é possível definir o VO como um segmento de reta, por dois pontos e, então, realizar o processo de com as duas MSRS, onde a primeira determina a VG (plano auxiliar paralelo à projeção horizontal do VO) do VO e a segunda MSR acumula o VO (plano auxiliar perpendicular à VG do VO). A Figura 16 mostra um exemplo de determinação de perspectiva axonométrica a partir de um vetor de observação dado. É possível notar que em termos das operações realizadas, o processo é exatamente o mesmo de determinação da VG de plano oblíquo. O que muda é a capacidade do usuário em determinar as direções adequadas dos planos nas MSR realizadas.

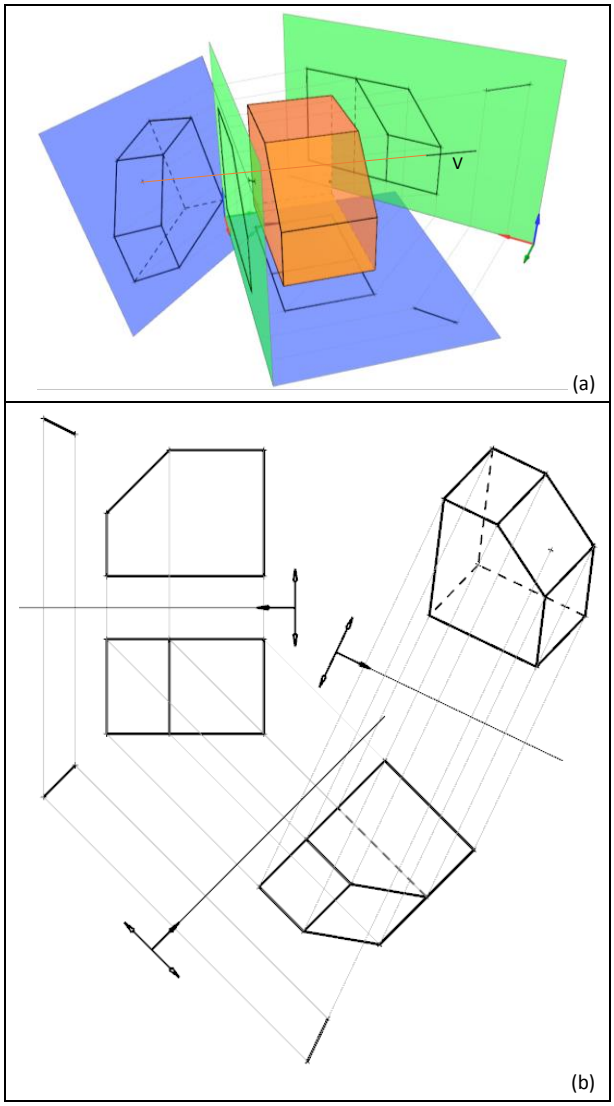


Figura 16: Determinação de perspectiva axonométrica a partir de uma vetor de observação (VO) em 3D (a) e em épura (b). Fonte: o autor.

A interface do HyperCAL^{3D} conta ainda com a possibilidade de obter a mesma perspectiva construída por MSRS no ambiente 3D através de um posicionamento da câmera adequado. Assim, se a câmera for posicionada com os mesmos parâmetros do VO, a janela 3D mostra o modelo sólido na perspectiva equivalente. Não se trata de uma épura e sim uma ferramenta de Computação Gráfica que pode ser utilizada para conferir o resultado feito através da GD.

3.5.2 Determinação das Vistas Ortográficas Principais

As vistas principais de um objeto são projeções alinhadas com os eixos do Sistema Referência e posicionadas segundo determinados critérios recomendados pela literatura e pelas normas do Desenho Técnico. As vistas principais são importantes, pois representam os objetos em vistas onde há verdadeiras grandezas de suas dimensões, possibilitando a sua construção física e permitindo intervenções ou

modificações da sua geometria na etapa de projeto. A vista principal, em geral, é a vista frontal, que caracteriza a posição de utilização ou que apresenta mais detalhes e um número menor de linhas invisíveis (NBR 10067, 1995, pg.4). As demais vistas são determinadas em função da vista principal.

Quando uma peça está representada em projeções que não correspondem às suas vistas principais, estas podem ser obtidas a partir de Mudanças de Sistema de Referência Sucessivas (MSRS), até que as vistas principais da peça sejam obtidas. O processo de obtenção de vistas principais por MSRS inicia pela escolha das direções que serão definidas como principais da peça e qual posição deverão ocupar em relação ao SR. Esta escolha está diretamente relacionada à escolha de vistas. Assim, ao definir como será a vista frontal (VF), é possível entender em que posição devem estar determinadas arestas e planos. No exemplo da Figura 17, após escolhida a VF e VS, é possível entender que as arestas paralelas à VF, que é oblíqua, deverão ficar posicionadas como retas de topo após o posicionamento das vistas. Além disso, a face ABCD deverá ficar horizontal quando as vistas escolhidas estiveram projetadas. A primeira MSR é realizada visando tornar a direção VF em horizontal (Figura 17-1). Isto é feito como uma etapa intermediária para tornar a direção VF em topo, pois toda reta de topo também é paralela ao PH. Portanto, esta primeira MSR é feita criando um novo plano horizontal paralelo à projeção frontal de VF, transformando esta aresta em horizontal no novo SR. O restante da projeção do sólido é construído normalmente.

A segunda MSRS é realizada para transformar a direção VF, que já está como horizontal na Vista Auxiliar Primária, em topo. Isto garante o posicionamento correto do objeto em relação ao SR para que a VF escolhida seja realmente reproduzida. Assim, uma nova linha de terra, correspondente a um novo PF, é traçada perpendicular à direção em VG de VF. Assim, a nova projeção é criada de forma que a direção VF se projeta como uma reta de topo. Esta nova projeção, uma VAS, consiste na vista frontal escolhida (Figura 17-2). No entanto, esta é a única Vista Principal até o momento. Ainda é necessário obter a vista superior da peça, para que a mesma fique totalmente alinhada com o SR.

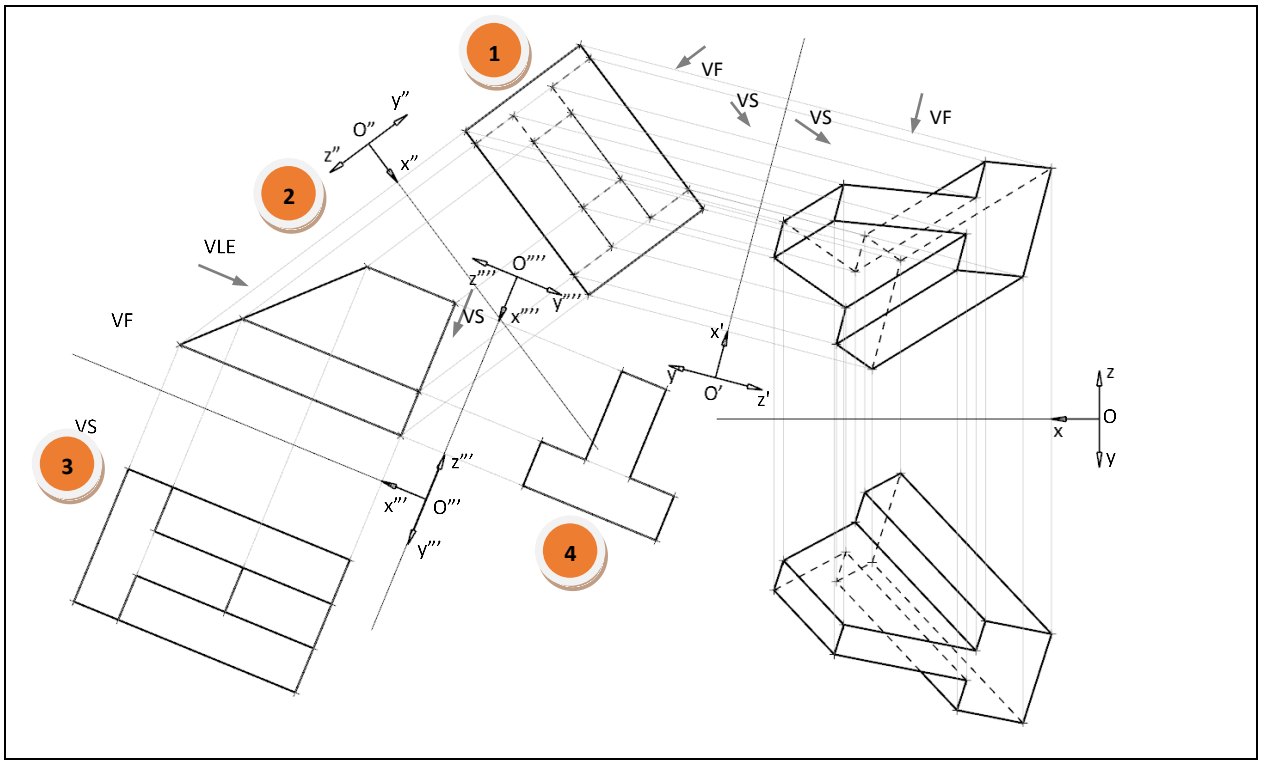


Figura 17: Obtenção das vistas ortográficas principais de um objeto utilizando MSRS no HyperCAL^{3D}. Fonte: o autor.

A obtenção da vista superior (VS) é feita através de uma terceira MSR sucessiva, criando uma nova Linha de Terra perpendicular à direção VS na segunda vista auxiliar. Desta forma, a face inferior fica posicionada como horizontal neste terceiro SR, o que é compatível com a vista superior escolhida. Assim, a terceira Vista Auxiliar corresponde à VS e a segunda corresponde à VF, as vistas principais da peça (Figura 17-3). A partir destas vistas, é possível reconhecer a forma e medir largura, altura e profundidade do objeto diretamente sobre as projeções.

A partir das Vistas Principais geradas, outras vistas podem ser criadas para auxiliar a visualização e o projeto. Um exemplo disto seria a obtenção de uma vista lateral esquerda. Neste caso, a vista auxiliar é feita com base na segunda vista auxiliar sucessiva (Figura 17-2), utilizando a direção correspondente à vista lateral esquerda (VLE), gerando a quarta vista auxiliar da peça, que corresponde à VLE (Figura 17-4). Assim, o quadro com três vistas principais do objeto foi obtido por uma série de vistas auxiliares.

Todo este processo é realizado em de forma interativas e em 3D no HyperCAL^{3D} (Figura 18), onde o usuário/estudante pode acompanhar, testar e entender o porquê de cada uma das MSR, ao mesmo tempo que acompanha a construção em épura (Figura 17).

O exemplo da Figura 17 é uma amostra do potencial do HyperCAL^{3D} como uma ferramenta de apoio ao ensino dos conceitos da Geometria Descritiva. Mas, assim como nos exemplos anteriores, o Programa não traz uma resposta automática e pronta para o estudante. É fundamental, para o seu uso, o conhecimento dos conceitos de GD. Assim, o HyperCAL^{3D} funciona como uma ferramenta de projeto assistido, que auxilia o processo de aprendizagem da GD.

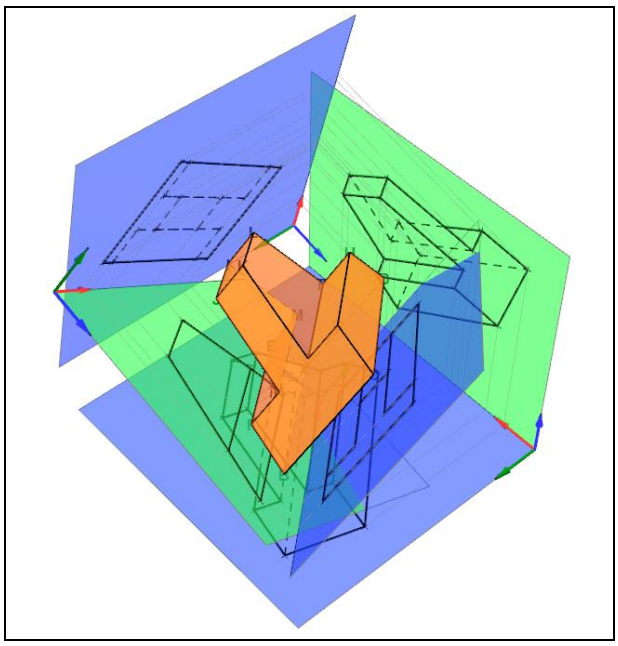


Figura 18: Obtenção das vistas ortográficas pricipais de um sólido por MSR sucessivas. Fonte: o autor.

4. CONSIDERAÇÕES FINAIS

O HyperCAL^{3D} é um programa computacional baseado em um conjunto de algoritmos fundamentados em geometria vetorial implementados em um ambiente gráfico tridimensional interativo, os quais reproduzem a metodologia e os procedimentos gráficos utilizados na Geometria Descritiva. O resultado é uma poderosa ferramenta, sem similar em nível nacional e internacional, para o ensino desta disciplina tão importante para as habilidades de visão e raciocínio espacial, fundamentais nas ações relacionadas ao projeto de artefatos nas diversas escalas.

Foram descritos os principais processos utilizados no desenvolvimento do Programa, que permitem a visualização, a manipulação de objetos sólidos e planos de projeção, ações que possibilitam o uso do HyperCAL^{3D} tanto por professores como por alunos em seus diferentes contextos de ensino e aprendizagem.

Os professores podem utilizar o Programa para preparar e testar exercícios, preparar material didático impresso e para apresentação e, ainda, para apresentar conteúdo de forma interativa para os alunos. Os estudantes podem utilizar o Programa para observar os processos em 3D e relacioná-los à épura buscando compreender a lógica por traz desta disciplina, que vem sendo ensinada por mais de dois séculos com base em axiomas e leis inquestionáveis. A observação dos processos em 3D e em épura, de acordo com a intenção próprio usuário/estudante, permite uma análise crítica e uma reflexão por parte destes, o que vai ao encontro da construção do próprio conhecimento.

O programa vem sendo utilizado desde 2006 em suas primeiras versões em sala de aula e a experiência tem sido extremamente positiva. Já é possível observar alunos em sala de aula com notebooks para testar seus projetos de forma virtual com o HyperCAL^{3D} antes de colocar no papel. Isto permite planejamento e a busca pela compreensão do que será feito, o que é fundamental para estudantes de Engenharia, Arquitetura e Design, que trabalham com projeto de artefatos. Entre os professores a receptividade tem sido boa, pois o uso do Programa permite reduzir a parte expositiva das aulas, resultando em um tempo maior para atividades práticas. A parte expositiva fica menor, mas mesmo assim é possível um aprofundamento maior pela qualidade do processo.

O programa está em sua versão 5.5 e continua em desenvolvimento e novas funcionalidades vêm sendo implementadas e testada, buscando a qualidade do ensino da Geometria Descritiva.

5. REFERÊNCIAS

[1] AGOSTON, M. K., 2005. *Computer Graphics and Geometric Modelling: Implementation & Algorithms.* s.l.:Springer.

[2] CHEN, D. M., 2000. Application of 3D CAD for Basic Geometric Elements in Descriptive Geometry. *Engineering Design Graphics Journal*, Volume 64: 01, pp. 10 - 17.

[3] HAWK, M. C., 1962. *Theory and Problems of Descriptive Geometry.* New York: Schaum Publishing Co..

[4] Jr, F. M. C., 1998. The Need(?) for Descriptive Geometry in a World of 3D Modeling. *Engineering Design Graphics Journal*, Volume 62: 03, pp. 4 - 8.

[5] PLAVJANIK, P., 1999. *Descriptive geometry.* [Online] Available at: http://dg.vidivici.cz/dg/dge.html [Acesso em 1 12 2012].

[6] RODRIGUES, Á. J., HAGUENAUER, C. J. & CUNHA, G. G., 2007. *A Realidade Aumentada no Ensino da Geometria Descritiva.* Curitiba, s.n.

[7] ROVIDA, E., 2013. *Machines and Signs - A History of the Drawing of Machines.* Amsterdam: Springer Netherlands.

[8] SANTOS, E. T., BARROS, L. N. D. & VALENTE, V. C. P. N., 2007. *Projetando uma Ontologia de Geometria Descritiva.* Curitiba, s.n., pp. 918 - 928.

[9] SCHNEIDER, P. & EBERLY, D. H., 2003. *Geometric Tools for Computer Graphics.* s.l.:Morgan Kaufmann.

[10] SEABRA, R. D. & SANTOS, E. T., 2013. Developing the Spatial Visualization Ability with a Virtual Reality Tool for Teaching Descriptive Geometry: a Brazilian Experience. *Journal for Geometry and Graphics*, Volume 17, pp. 101 - 117.

[11] TEIXEIRA, F. G., 2003. *Modelamento Paramétrico e Geração de Malha em Superfícies para Aplicações em Engenharia,* Porto Alegre: s.n.

[12] TEIXEIRA, F. G., SILVA, R. P. D. & SILVA, T. L. K. d., 1999. *A Learning Environment for Teaching of Descriptive Geometry.* Ostrava - Prague, s.n.

[13] TEIXEIRA, F. G., SILVA, R. P. D., SILVA, T. L. K. d. & Hoffman, A. T., 2006. *The Descriptive Geometry Education Through the Design-Based Learning.* Salvador, s.n.

[14] WALKER, P. R., 2003. *The Feud That Sparked the Renaissance: How Brunelleschi and Ghiberti Changed the Art World.* s.l.:Harper Perennial.

[15] WATT, A., 1993. *Computer Graphics.* s.l.:s.n.

[16] WRIGHT, R. S., Lipchak, B. & Haemel, N., 2007. *OpenGL SuperBible: Comprehensive Tutorial and Reference.* 4th Edition ed. s.l.:Addison-Wesley.