



ESCOLA DE ENGENHARIA
FACULDADE DE ARQUITETURA
PROGRAMA DE PÓS-GRADUAÇÃO EM DESIGN

José Ernesto Bueno Wills

USO DE TECNOLOGIAS DIGITAIS NAS ETAPAS INICIAIS DE
PROJETO ARQUITETÔNICO

Porto Alegre
2014



ESCOLA DE ENGENHARIA
FACULDADE DE ARQUITETURA
PROGRAMA DE PÓS-GRADUAÇÃO EM DESIGN

José Ernesto Bueno Wills

USO DE TECNOLOGIAS DIGITAIS NAS ETAPAS INICIAIS DE PROJETO ARQUITETÔNICO

Dissertação submetida ao Programa de Pós-Graduação
em Design da Universidade Federal do Rio Grande do
Sul para a obtenção do Grau de Mestre em Design.

Orientador: Prof. Dr. Benamy Turkienicz

Porto Alegre

2014

CIP - Catalogação na Publicação

Wills, José Ernesto Bueno

Uso de tecnologias digitais nas etapas iniciais
de projeto arquitetônico / José Ernesto Bueno Wills. -
- 2014.
120 f.

Orientador: Benamy Turkienicz.

Dissertação (Mestrado) -- Universidade Federal do
Rio Grande do Sul, Faculdade de Arquitetura,
Programa de Pós-Graduação em Design, Porto Alegre, BR-
RS, 2014.

1. projeto arquitetônico. 2. cognição de projeto.
3. representação arquitetônica. 4. interpretação de
sketches. 5. interação humano-computador. I.
Turkienicz, Benamy, orient. II. Título.

José Ernesto Bueno Wills

USO DE TECNOLOGIAS DIGITAIS NAS ETAPAS INICIAIS DE PROJETO ARQUITETÔNICO

Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Design, e aprovada em sua forma final pelo Programa de Pós-Graduação em Design da UFRGS.

Porto Alegre, 19 de dezembro de 2014.

Prof. Dr. Fábio Gonçalves Teixeira
Coordenador do Programa de Pós-Graduação em Design da UFRGS

Banca Examinadora:

Prof. Dr. Benamy Turkienicz
Orientador
Universidade Federal do Rio Grande do Sul

Prof.^a Dr.^a Luciana Porcher Nedel
Universidade Federal do Rio Grande do Sul

Prof. Dr. Fábio Gonçalves Teixeira
Universidade Federal do Rio Grande do Sul

Prof. Dr. Joaquim Armando Pires Jorge
Instituto Superior Técnico, Universidade de Lisboa

AGRADECIMENTOS

Este trabalho não teria podido ser desenvolvido sem o apoio incondicional dos meus pais, Euridys e Emilio e da minha esposa, Rita.

Estou muito agradecido ao Professor Benamy Turkienicz. A sua orientação foi determinante, tanto em forma como em conteúdo, além de me ensinar o verdadeiro rigor e metodologia de pesquisa acadêmica. Agradeço ao Laboratório para Simulação e Modelagem em Arquitetura e Urbanismo – SimmLab, pelo ambiente de trabalho, a camaraderia e o apoio. Especialmente agradeço a Renato Silveira pelos valiosos conselhos sobre computação, a Vanessa Gallardo e Lennart Poehls pelo feedback sobre projeto e desempenho ambiental e a Eduardo Carneiro, pela ajuda no trabalho braçal. Agradeço à Professora Luciana Nedel pelas assessorias sobre interação humano-computador.

Agradeço também às pessoas que retornaram para mim informações atualizadas sobre tecnologia e projeto: Verônica Natividade, Pedro Veloso e Leandro Michelena; e a Bento Pimentel e Felix Bressan, que, em momentos diferentes, contribuíram a enriquecer a minha visão sobre projetos e também me acolheram em Porto Alegre.

Este trabalho foi realizado com apoio parcial da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, do Ministério da Educação – CAPES – Brasil; e com apoio parcial da Fundação de Apoio da Universidade Federal do Rio Grande do Sul – FAURGS.

RESUMO

Em projetos arquitetônicos, lápis e papel continuam sendo os instrumentos mais usados para elaborar desenhos à mão livre para dar suporte às atividades das etapas iniciais de projeto. Ferramentas digitais convencionalmente utilizadas por arquitetos carecem de funcionalidade apropriada e não oferecem interfaces amigáveis para os estágios iniciais de concepção arquitetônica. A solução para estes problemas fundamenta-se em dois apoios tecnológicos: (i) em dispositivos de hardware já disponíveis que podem ajudar projetistas a expressar suas ideias através de sketches à mão livre; e (ii) em princípios de funcionalidade presentes em softwares experimentais os quais, combinados com hardware apropriados, poderiam oferecer interações humano-computador amigáveis e intuitivas nas etapas iniciais de projetos arquitetônicos. Esta pesquisa revisa e sintetiza literatura relevante à esta hipótese, especialmente quanto às formas como projetistas de arquitetura interagem com os computadores, como dispositivos de entrada restringem possíveis interações e como diferentes funcionalidades podem ser exploradas através dessas interações. A revisão do estado da arte dos softwares experimentais de modelagem baseada em sketch para projetos arquitetônicos permitiu descrever em detalhe as funcionalidades oferecidas e argumentar que, se combinadas, essas funcionalidades forneceriam um apoio mais adequado para as etapas iniciais de projeto arquitetônico. Cenários de uso são desenvolvidos para explorar tais combinações e demonstrar possibilidades atuais e futuras das ferramentas digitais durante a fase de concepção do projeto arquitetônico.

Palavras-chave: projeto arquitetônico, cognição de projeto, representação arquitetônica, interpretação de sketches, interação humano-computador.

ABSTRACT

In architectural design, pencil and paper remain the most used media to create freehand drawings to support the activities of early designing stages. Digital tools conventionally used by architects lack of appropriated functionality and do not offer friendly interfaces for early stages of architectural designing. The solutions for these problems are twofold: (i) hardware already available can help designers to express their first ideas by freehand sketches; and (ii) functionality principles are present in experimental software and, combined with appropriate hardware, could successfully provide a friendly and intuitive human-computer interaction in the early stages of architectural designing. This research revises and synthesizes literature relevant to this hypothesis, taking special attention to the way architectural designers interact with computers, how input devices constrain possible interactions and how functionalities can be explored through these interactions. The update to the state-of-the-art studies in experimental, sketch-based modeling software for architectural design allowed detailed descriptions of the functionality they offer and allowed to argue that, combined, this functionality would provide a more adequate support for initial stages of architectural designing. Scenarios of use are developed, exploring such combination and demonstrating current and future possibilities for digital tools to use during initial stages of architectural designing.

Keywords: architectural design, design cognition, architectural representation, sketch interpretation, human-computer interaction.

LISTA DE ILUSTRAÇÕES

	Pág.
Figura 1: Reconhecimento de uma forma emergente num sketch arquitetônico em planta.	5
Figura 2: Sketches em corte que ilustram as consequências arquitetônicas de interpretações de uma forma emergente.	5
Figura 3: Perspectivas mostrando modelos 3D de diferentes interpretações de sketches.	5
Figura 4: Transição de mídias no processo de projeto de arquitetura.	20
Figura 5: Paradigmas de IHC segundo Heim.	22
Figura 6: Mapeamento dos paradigmas de IHC relevantes.	23
Figura 7: Protocolo para marcar pontos de divisão numa linha em Autodesk AutoCAD.	26
Figura 8: Duplicação de um gráfico por meio de um gesto que combina entrada por caneta e <i>touch</i> , usando Manual Deskterity.	36
Figura 9: Usuário desenhando um sketch com uma caneta <i>stylus</i> num computador <i>tablet</i> como entrada de modelagem no software GREGROSS.	37
Figura 10: Capturas do teste do Structural Sketcher.	44
Figura 11: Janelas do EsQUIsE interpretando os sketches de um arquiteto.	45
Figura 12: Interpretação de portas, janelas e mobiliário desenhado em NEMo.	46
Figura 13: Processo de construção geométrica em SESAME.	46
Figura 14: Processo de modelagem baseada em sketch da Casa da Cascata em SmallBoom.	47
Figura 15: Sequência de sketches que se sobrepõem em Mental Canvas para descrever uma casa.	47
Figura 16: Design da interface gráfica de usuário do aplicativo de Kang, Kim <i>et al.</i> (2013), sinalizando os elementos de controle dos quatro modos de operação.	50
Figura 17: Esquema de adição de sensores para captação de entrada multimodal num <i>tabletop</i> ótico.	52
Figura 18: Utilização do ShapeShop Multitouch no <i>tabletop</i> ótico adaptado.	52
Figura 19: Interface do REFER na reconstrução automática de um sketch em perspectiva cavaleira por meio de algoritmos de otimização.	55
Figura 20: Janela de configuração dos parâmetros da otimização Hill Climbing para reconstrução automática em REFER.	56

Figura 21: Janela de detalhes de configuração da otimização Hill Climbing para reconstrução 3D no REFER.	56
Figura 22: Janela de ajuda sobre os gestos reconhecidos em REFER e CIGRO.	57
Figura 23: Processo de modelagem em CIGRO a partir do reconhecimento e interpretação em tempo real.	58
Figura 24: Sequência de desenho paramétrico por restrições em ParSketch.	59
Figura 25: Sequência de modelagem em GEGROSS. Exemplo que combina gestos de revolução e extrusão.	60
Figura 26: Exemplo de sessão de projeção por reconstrução automática de sketches e simulação de desempenho estrutural em duas iterações.	62
Figura 27: <i>Storyboard 1</i> : traços iniciais de um sketch reconhecidos automaticamente como formas geométricas.	75
Figura 28: <i>Storyboard 1</i> : Ativação da vista em corte, redesenho de coberturas e verificação das modificações em perspectiva.	77
Figura 29: <i>Storyboard 1</i> : O desenho do rótulo Norte define orientação e permite simulações de desempenho ambiental.	78
Figura 30: <i>Storyboard 1</i> : Rotacionando e orbitando o modelo recalcula-se a simulação de Sombreamento.	78
Figura 31: <i>Storyboard 1</i> : Composição de fachadas através do desenho e transformação avançada de um módulo, utilizando gestos bimanuais.	79
Figura 32: <i>Storyboard 2</i> : Navegação <i>multi-touch</i> pelo mapa terráqueo para definição de localização do projeto.	80
Figura 33: <i>Storyboard 2</i> : Desenho de um sketch de pensamento sendo adquirido pelo software sem interrupções.	81
Figura 34: <i>Storyboard 2</i> : Edição manual de uma extrusão, para modificar a altura de um edifício reconstruído.	82
Figura 35: <i>Storyboard 2</i> : Simulação de sombreamento (a) e de iluminância (b) sobre os edifícios e o espaço urbano sendo projetado.	82
Figura 36: Princípios de funcionalidade necessários em softwares de apoio nas etapas iniciais de projeto arquitetônico.	92

LISTA DE TABELAS

	Pág.
Tabela 1: Técnicas, ferramentas e suportes comumente associados a tipos de representação em projetos de arquitetura.	19
Tabela 2: Características de interação dos softwares estudados	48
Tabela 3: Funcionalidades para desenho, construção e visualização do modelo nos softwares estudados.	53
Tabela 4: Características de computação dos softwares experimentais estudados.	63
Tabela 5: Linguagens de programação empregadas no estado da arte e compatibilidade oferecida.	67
Tabela 6: Gestos de navegação propostos.	72
Tabela 7: Gestos propostos para transformações e edições avançadas.	73
Tabela 8: Interações e processamento segundo processos cognitivos, sugeridos nos cenários de uso.	84
Tabela 9: Funcionalidades necessárias para suportar as atividades de cada cenário de uso.	87

SUMÁRIO

Introdução.....	1
Representações relevantes nas etapas iniciais do projeto de arquitetura.....	6
Transferência do papel para o ambiente digital.....	8
Interação dos dispositivos de entrada para projetos de arquitetura.....	8
<i>Affordance</i> a partir das simulações de desempenho.....	10
Disponibilidade das tecnologias de simulação de desempenho e entrada por caneta e <i>multi-touch</i> para arquitetos.....	11
Caracterização do problema da pesquisa.....	13
1 Fundamentação teórica.....	17
1.1 Técnicas e ferramentas relevantes em projeto.....	17
1.1.1 Técnicas de desenho mais comuns nas etapas iniciais de projeto.....	17
1.1.2 Tipos de sketch segundo a atividade projetual.....	20
1.2 Interação entre o projetista e o computador.....	21
1.2.1 Paradigmas de interação humano-computador.....	21
1.2.2 Evolução da interação por teclado e mouse.....	24
1.2.3 Prejuízos da utilização de ferramentas CAD nas etapas iniciais de projeto.....	27
1.2.4 Interação das simulações de desempenho.....	29
1.2.5 Evolução da interação por caneta e <i>touch</i>	31
1.2.6 Interação multimodal relevante para o processo de projeto.....	35
1.2.7 Evolução da modelagem baseada em sketch.....	37
1.3 Adoção de tecnologias de caneta e <i>multi-touch</i>	38
1.3.1 Adoção de canetas e <i>multi-touch</i> em outras disciplinas criativas.....	38
1.3.2 Falta de adoção em projetos de arquitetura e urbanismo.....	39
2 Estado da arte.....	44
2.1 Configuração geométrica testada.....	44
2.2 Modos de entrada.....	48
2.2.1 Entrada por caneta <i>stylus</i>	49
2.2.2 Entrada <i>multi-touch</i>	49
2.2.3 Entrada multimodal.....	51
2.3 Funcionalidades de editoração de geometria.....	52
2.3.1 Reconstrução automática.....	54
2.3.2 Reconstrução gestual.....	57
2.3.3 Sketches em 2D ou 3D.....	60
2.3.4 Simulação de desempenho.....	61

2.3.5	Computação e interoperabilidade do modelo	62
2.4	Potencial de reaproveitamento e integração das funcionalidades	65
3	Funcionalidades necessárias para apoiar as etapas iniciais de projeto arquitetônico	69
3.1	Descrição das funcionalidades de software relevantes	69
3.1.1	Interação arquiteto-computador	70
3.1.2	Definição de vocabulário de gestos	71
3.1.3	Processamento geométrico	73
3.2	Estabelecimento de cenários de utilização	74
3.2.1	Cenário 1	75
3.2.2	Cenário 2	79
3.2.3	Síntese de funcionalidades sugeridas nos cenários de utilização	83
4	Conclusões	91
4.1	Princípios de funcionalidade para apoiar as etapas iniciais de projeto arquitetônico	92
4.1.1	Interação bimanual multimodal	92
4.1.2	Modelagem baseada em sketch	93
4.1.3	Simulações preliminares de desempenho	94
4.1.4	Transição entre etapas de projeto	94
4.2	Desenvolvimento futuro	95
5	Referências	97
	Glossário	106

INTRODUÇÃO

Diferentemente da atuação profissional do designer, a pesquisa em design não está voltada diretamente a produtos específicos nem ao usuário final, mas ao estudo dos processos que levam a produção dos objetos que chegam ao usuário final. Processos desenvolvidos por designers, para cada um dos projetos que dão origem a estes objetos, devem ser avaliados para evidenciar falhas e acertos e, assim, definir diretrizes de desenvolvimento das duas atividades, a da pesquisa e a da projeção (FASTE e FASTE, 2012).

A dualidade acima descrita guarda estreita relação com o papel tecnológico do design¹, já que a pesquisa em design consegue descrever a fenomenologia dos processos de projeção e relacioná-la com a ontologia das tecnologias associadas à disciplina de projetos. Esta relação conduz a discussões sobre (i) o uso que o projetista faz da tecnologia; (ii) questionamentos sobre até que ponto a tecnologia está servindo de apoio ou produzindo uma limitação nos processos e produtos da arquitetura e do design (STEELE, 2001) (LAWSON, 2004); e (iii) o seu impacto na realização dos objetivos principais do design perante os usuários-alvo destes produtos de design e, em última instância, perante a sociedade (BÜRDEK, 2005). Assim, compete também ao pesquisador em design estudar a atividade do projetista na composição de elementos e intenções de projeto, levando em conta a grande quantidade de fatores a partir dos quais são formulados os problemas de projeto.

Os problemas de projeto podem ser agrupados, segundo a estrutura de solução, em quatro tipos (MATTHEWS, 1998): (1) problemas técnicos: problemas considerados como bem definidos cuja solução depende somente de uma sequência linear de passos técnicos (ex.: problemas matemáticos); (2) problemas procedurais: aqueles nos quais os procedimentos possíveis direcionam-se para uma solução linear (ex.: interoperabilidade, qualidade segundo normas); (3) problemas fechados: aqueles com uma formulação curta, mas com uma complexidade de soluções não aparente, estruturada em camadas que só exibem a sua complexidade técnica ou procedural quando abordadas (ex.: confiabilidade, desenvolvimento de novos produtos); e (4) problemas aninhados: aqueles que possuem vários níveis de soluções

¹ Das quatro tarefas principais que, segundo Bürdek (2005), o design do século XXI deve cumprir, as duas primeiras se referem ao acompanhamento do avanço da tecnologia para viabilizar a disseminação e o uso de produtos de software e hardware. Para acadêmicos seminiais da teoria do design como Bürdek (*ibid.*), o avanço do design deve ser feito primeiramente na direção tecnológica, centrada no usuário.

parciais que, além de serem fechados, no momento da sua abordagem, levam à formulação de outros problemas (ex.: termodinâmica, gestão, questões criativas) (*ibid.*).

Nas áreas de arquitetura, urbanismo e design de produto, os problemas de projeto são geralmente do tipo “aninhado” (*op. cit.*). A solução é construída iterativamente através de sequências de representações (gráficas e/ou físicas) que abordam progressivamente problemas dentro de problemas. O ato de projetar se materializa em representações visuais que permitem a discussão, viabilização e execução, com repercussões significativas para o usuário e o seu contexto (GERO, 1990). Por esta razão, a representação de projeto deve ser um instrumento comunicativo, claro e ágil.

Os problemas de projeto também podem ser identificados, pela natureza da sua formulação, como problemas mal definidos. Os problemas bem definidos (*well-defined problems*), característicos nas ciências exatas, são aqueles nos quais todas as tarefas que compõem a solução têm uma formulação operativa. Em contraposição, nos problemas mal definidos (*ill-defined problems* ou *wicked problems*), como os de arquitetura e design, parte da especificação do problema é desconhecida (EASTMAN, 1969) (LAWSON, 2005). Nas etapas iniciais de projeto, ao fazer uma abordagem geral, o projetista se vê na necessidade de definir alguns aspectos e deixar outros parcialmente definidos ou difusos, dando espaço para futuras interpretações e para a construção de argumentos. Uma vez que a evolução do projeto avança nos subníveis dos problemas aninhados, trabalha-se num contexto melhor direcionado pela prévia abordagem de problemas gerais e numa escala mais próxima, permitindo uma melhor definição destes problemas específicos (MOREIRA e KOWALTOWSKI, 2009).

A natureza mal definida dos problemas de projeto é evidente em tarefas como o planejamento espacial ou *layout*, onde não se delimita formalmente o domínio do problema nem se tem uma expressão precisa sobre o que define uma solução aceitável (EASTMAN, 1969). Poderão existir vários *layouts* potencialmente aceitáveis, uns mais coerentes do que outros com relação à intencionalidade do projetista, que os seleciona seguindo uma valoração pessoal baseada em conhecimento explícito e implícito (LAWSON, 2005). O conhecimento explícito é o conhecimento articulado, que é comunicado formalmente através de uma mídia escrita (ex.: parâmetros antropométricos, catálogo de mobiliário). Conhecimento implícito e conhecimento tácito são aqueles que, sem estar articulados, estão presentes no projetista (ex.: hierarquia de usos, critérios de ordenamento de espaços) e, para sua comunicação, são plausíveis de articulação através de externalizações (NICKOLS, 2002). Estas externalizações podem ser feitas através da escrita, mas, em arquitetura, urbanismo e design, o projetista recorre principalmente à documentação visual por meio de desenhos para externalizar a sua valoração

com respeito ao projeto e registrar a sua tomada de decisões (OXMAN, 1997). O projetista aplica uma diversidade de conhecimentos implícitos no momento da elaboração de projetos: por exemplo, a sua formação e experiência dotam ao projetista de conhecimento tácito sobre noções de física aplicada (ex.: física estática aplicada a estruturas de edificações; termodinâmica aplicada ao conforto ambiental) (WOO, CLAYTON *et al.*, 2004). Na tomada das primeiras decisões de projeto, o arquiteto não elabora cálculos numéricos, se vale deste conhecimento para inferir soluções no momento de externalizá-las por meio dos desenhos.

O ato de desenhar, como atividade projetual, é uma transação visual e mental, uma conversação que o projetista estabelece consigo mesmo (SCHÖN *apud* SUWA e TVERSKY, 1997) (OXMAN, 1997). O processo de ideação do projeto é conformado pelo ciclo reflexivo: desenhar, inspecionar, revisar. Os avanços de projeto são atingidos graças a sequências de ações do projetista que resultam em transformações da representação do projeto (OXMAN, 1997). Para atingir fluidez no avanço através do ciclo reflexivo, o projetista deve ter a habilidade de transformar conhecimento implícito em estruturas representacionais, e fazê-lo numa velocidade rápida o suficiente para que as novas modificações e alterações reflitam a interpretação do revisado, criando uma continuidade entre a mente e o desenho em elaboração (LAWSON, 2005).

Para estabelecer esta conexão entre a mente e o desenho em progresso, o projetista faz uso da sua **inteligência visual** (HOFFMAN, 1998) para fazer associações entre as formas desenhadas de maneira diferente à adotada num primeiro momento (GERO e YAN, 1993). A inteligência visual é um dos principais tipos de inteligência humana conectada com partes importantes do cérebro² (GARDNER, 1985). Ela ocupa perto da metade da atividade do córtex, provendo da habilidade de “ver” coisas com a mente, interpretadas a partir das imagens inicialmente registradas através dos olhos (HOFFMAN, 1998). A grande quantidade de recursos cognitivos atribuídos à inteligência visual garante um grande potencial de desenvolvimento desta inteligência (*ibid.*). Esta inteligência é especialmente desenvolvida em arquitetos e designers, que exploram este potencial com o treinamento visual, conformado pela experiência da sua formação e exercício profissional. Isto lhes habilita a encontrar formas e padrões emergentes que permitem a reinterpretar de um problema de projeto. O processo de formação destas formas emergentes, ou **emergência visual**, faz explícitas as características e relações implícitas entre elementos originados por outras relações (GERO e YAN, 1993).

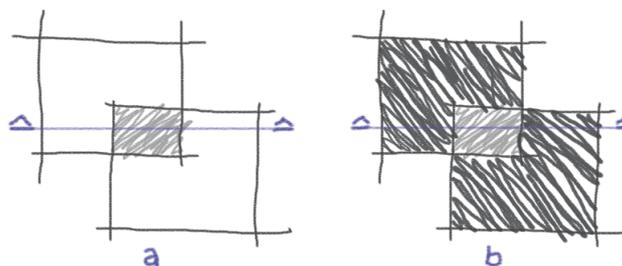
² Segundo a teoria das múltiplas inteligências de Gardner (1985), diferentes partes do cérebro estão conectadas com diferentes tipos de inteligência, identificando sete: verbal, lógica, visual, musical, física-cinética, comunicativa e cognitiva (*ibid.*).

Diferentemente do conceito de emergência tratado em outras áreas do conhecimento³, a emergência visual não é necessariamente imprevisível, pode ser prevista pelo projetista utilizando sua inteligência visual para adiantar o processo de solução (OXMAN, 2002). No contexto dos projetos de arquitetura, esta emergência permite a interpretação destas formas como um ou outro elemento arquitetônico. Por exemplo, na Figura 1 (em azul, marcações posteriores), o projetista reconhece, na sobreposição de dois retângulos, uma forma retangular emergente. Ela pode ser interpretada como uma área sombreada na sobreposição de volumes de base retangular (Figura 1a), ou como um vazio no encontro entre dois volumes (Figura 1b). O projetista consegue interpretar estas formas emergentes como elementos arquitetônicos com diferentes propriedades espaciais que acontecem numa terceira dimensão não representada. Na necessidade de externalizar estas propriedades, o projetista pode elaborar outros sketches que as ilustrem (Figura 2), e assim pode argumentar de uma maneira menos ambígua que a primeira interpretação definiria um espaço interno sem incidência direta da luz do sol (Figura 3a), enquanto a segunda interpretação definiria um espaço interno com incidência direta da luz do sol (Figura 3b). Ao aproveitar esta habilidade, o projetista abre a possibilidade de avançar numa solução que aborde, através do redesenho, outros problemas até então não abordados (*ibid.*).

A sequência de desenhos atua como um registro do processo de raciocínio que o projetista constrói a partir da inspeção e revisão do desenho anterior, que o leva à transição de uma representação à outra (OXMAN, 1997). A “re-representação”, ou sequência de representações produto deste redesenho, pode resultar em propostas de novidade e diversidade inesperada (*ibid.*). Ao inspecionar a sua sequência de desenhos, o projetista vê relações e características que não eram claras até então, o que sugere maneiras de revisar as ideias (SUWA e TVERSKY, 1997). Através desta “re-representação”, o novo é construído a partir do existente (OXMAN, 1997).

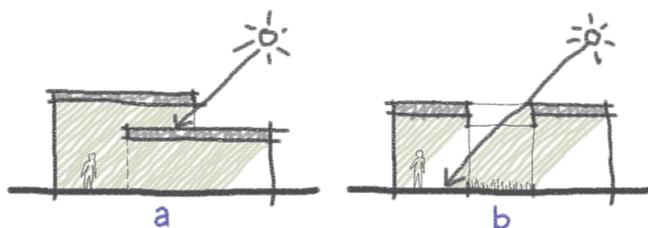
³ O conceito de emergência é conhecido em outras áreas, como nas ciências da complexidade, como um fenômeno auto-organizativo relacionado à imprevisibilidade ou à aleatoriedade. O conceito de emergência visual, ou emergência de formas (*shape emergence*), aqui citado tem sido estudado em relação a uma intencionalidade criativa presente no design, na arquitetura e nas artes (GERO e YAN, 1993) (OXMAN, 2002).

Figura 1: Reconhecimento de uma forma emergente num sketch arquitetônico em planta.



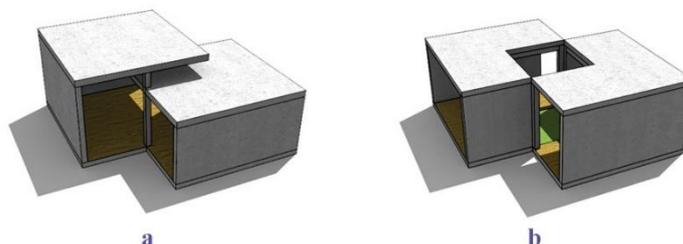
Fonte: o autor.

Figura 2: Sketches em corte que ilustram as consequências arquitetônicas de interpretações de uma forma emergente.



Fonte: o autor.

Figura 3: Perspectivas mostrando modelos 3D de diferentes interpretações de sketches.



Fonte: o autor.

Ao ficarem focados no pensamento e representação de soluções de projeto dentro do ciclo reflexivo, os projetistas não costumam explicitar quais os conhecimentos e habilidades usadas no momento de projetar, nem como estas habilidades são usadas (SUWA, PURCELL e GERO, 1998). Coordenam implicitamente uma série de métodos pessoais a partir da experiência de projetar através da **percepção construtiva** (SUWA, 2003), uma habilidade, um mecanismo da mente que faz a pessoa treinada coordenar ações cognitivas de reorganização perceptiva (como através da emergência visual) e de geração conceitual (através da re-representação)⁴. Com esta habilidade, os projetistas aperfeiçoam os métodos pessoais com os

⁴ A explicação sobre a percepção construtiva aqui apresentada é uma citação indireta de Suwa (2003). As frases entre parêntesis integram o nosso entendimento sobre a emergência visual (OXMAN, 2002) e a re-representação (OXMAN, 1997), numa relação intrínseca com a percepção construtiva, mesmo que não esteja explicitada em Suwa (2003).

quais conseguem boas soluções de projeto e, pela experiência prática, ficam mais rápidos e melhor conectados com a intuição (*ibid.*).

REPRESENTAÇÕES RELEVANTES NAS ETAPAS INICIAIS DO PROJETO DE ARQUITETURA

O projetista utiliza habilidades de emergência visual e percepção construtiva para avançar no processo de projeto de maneiras diferentes em cada etapa de projeto. A diferença no emprego das habilidades está em estreita relação com a mudança da natureza mal definida dos problemas ao longo do processo de projeto. As etapas assumem denominações específicas segundo a resolução dos projetos, mas todas guardam a mesma relação de progressão desde a mais difusa até a mais precisa. Em design de produto estas etapas são denominadas rascunho, design conceitual, detalhamento e design de produção (BRUCE e BESSANT, 2002); enquanto no projeto arquitetônico denominam-se partido arquitetônico, estudo preliminar, anteprojeto e projeto executivo (MOREIRA e KOWALTOWSKI, 2009).

Na representação arquitetônica, os tipos de representação tradicionalmente relevantes nas tomadas de decisões são o **sketch** e o **desenho técnico**. Nas últimas décadas tem se tornado relevante o **modelo 3D** (MONEDERO, 2000) e as **simulações de desempenho** (NOME e QUEIROZ, 2013). Estes quatro tipos de representação são produzidos empregando diversas técnicas e ferramentas e têm pesos e funções diferentes através das etapas de projeto (*ibid.*) como, a seguir, descrito.

Sketch: Para melhor conseguir agilidade de conexão produtiva – cognição-representação de ideias difusas –, o projetista faz um desenho informal composto por traços rápidos, às vezes rabiscados, com um baixo nível de precisão e um alto nível de expressividade, associados à construção de ideias incipientes e designs conceituais, assim como à anotação e revisão de outros desenhos. Este é o tipo de representação que em pesquisa em design denominamos **sketch**, e na arquitetura é tradicionalmente conhecido como *croquis*⁵ (FLORIO, 2010).

No desenvolvimento do projeto através das diferentes etapas, ao tornarem-se progressivamente mais precisos, os problemas podem ser solucionados com desenhos técnicos, elaborados com técnicas de CAD (DENZER e GARDZELEWSKI, 2011). Nas etapas iniciais, entretanto, os problemas são tão difusos, que o desenho a mão continua sendo a técnica mais empregada (CARPO, 2013). Pode-se, então, afirmar que o sketch é o tipo de representação mais

⁵ Ante o dilema de utilização das palavras *croquis* ou *sketch*, nesta dissertação se estabelece o uso de sketch por ser o termo preferido em pesquisa em design e em computação.

apropriado para abordar os problemas de projeto na sua fase inicial (COMPANY, CONTERO *et al.*, 2006) (FLORIO, 2010) (DENZER e GARDZELEWSKI, 2011).

O fluxo de trabalho mais representativo do arquiteto começa pela recollecção de dados contextuais, pela formulação dos problemas de projeto e pela elaboração de uma sequência de sketches que abordam progressivamente os problemas formulados (OXMAN, 1997). O projetista aproveita a sua habilidade de reorganização de formas emergentes (OXMAN, 2002), coordenando-as na geração conceitual (SUWA, 2003) das re-representações (OXMAN, 1997), definindo o partido arquitetônico.

As atividades das etapas iniciais de projeto, que definem o partido arquitetônico, acontecem tipicamente através de sketches desenhados à mão livre com lápis e papel (ASANOWICZ, 2003) (DENZER e GARDZELEWSKI, 2011). A ideia plasmada no papel externaliza, através de representações vagas ou difusas, a intenção do projetista na abordagem de problemas mal definidos.

Desenhos técnicos e modelagem 3D: Nas etapas avançadas, o projeto é representado em meio digital com: desenhos técnicos (MOREIRA e KOWALTOWSKI, 2009) tipicamente elaborados utilizando ferramentas computacionais como Autodesk AutoCAD; perspectivas utilizando softwares de modelagem em 3D como Trimble SkekcthUp (WINSTON, 2010); e simulações de desempenho utilizando softwares como Autodesk Ecotect (NOME e QUEIROZ, 2013).

Simulação de Desempenho: Os softwares de simulação de desempenho incluem recursos visuais que modificam temporariamente a aparência do modelo de maneira distinta segundo o fator ambiental a simular. Por exemplo, a simulação de vento desenha um campo de vetores no entorno, representando o fluxo do ar; enquanto a simulação de iluminação faz uma discretização do modelo, amostrando-o numa malha poligonal e colorindo a malha com uma variação cromática segundo a quantidade de luz incidente para cada vértice⁶.

Os softwares de CAD, modelagem 3D e simulações de desempenho, oferecem funcionalidades adicionais de desenho direcionadas para representações mais precisas que as produzidas com desenhos à mão livre. Para avançar nas etapas de projeto em meio digital, é necessário transferir as ideias do partido, plasmadas em papel, para o ambiente computacional e, ao fazê-lo, aumentar a resolução da informação, explicitando detalhes que no sketch podiam ser difusos ou até implícitos (YU e ZHANG, 2007).

⁶ Em simulações como a de iluminação natural, a malha é discretizada segundo uma amostragem (subdivisão a uma distância constante nas três dimensões, ou distância de amostragem) gerando uma densidade maior e mais constante de vértices, aos quais se atribuem variações de cores que definem a aparência das faces poligonais da malha: cores cálidas nas fases que representam regiões em torno a pontos mais iluminados; e cores frias nas de menor iluminação.

Transferência do papel para o ambiente digital

A transferência é normalmente feita por meio da digitalização e redesenho em CAD, em *layers* ou camadas superpostas acima da imagem escaneada (mapa de bits) do sketch (*overlay*); ou através da construção de um novo desenho CAD, iniciado em branco (sem uma prévia inserção de imagens), com base na observação e recriação das características do sketch (MASRY e LIPSON, 2005). Esta transferência consome um tempo de projeto no qual, pela interpretação, são revisadas decisões de projeto já tomadas (DENZER e GARDZELEWSKI, 2011) até que a representação atinja a definição apropriada para abordar os problemas das etapas mais avançadas de projeto (MOREIRA e KOWALTOWSKI, 2009). O esforço para transferir decisões de projeto da mídia em papel para a mídia digital consome o tempo de desenho da mídia digital, tornando o processo demorado. Isto retarda o processo projetual tornando-o, nesta fase, pouco eficiente (COMPANY, CONTERO *et al.*, 2006).

A ineficiência na transferência entre mídias poderia ser evitada, ao elaborar todas as etapas numa só mídia. Partindo do requerimento de trabalhar em mídia digital nas etapas avançadas de projeto, é uma consequência natural considerar as tecnologias digitais para o desenvolvimento do projeto desde as etapas iniciais (DENZER e GARDZELEWSKI, 2011). Existem softwares para usuários finais capazes de permitir que projetistas empreguem técnicas de desenho à mão livre para a elaboração de sketches característicos das etapas iniciais de projeto, desde que sejam instrumentados por dispositivos de entrada apropriados, como canetas *stylus* (VIZIOLI, LANCHI e CASTRAL, 2013). Entretanto, trata-se de softwares de edição de imagens em mapa de bits. O formato de imagem gerado nestes softwares não é compatível com o formato requerido pelos softwares CAD, de modelos vetoriais. A vetorização necessária para trabalhar nos softwares CAD também traz os problemas de transferência acima mencionados.

Interação dos dispositivos de entrada para projetos de arquitetura

Até recentemente, tecnologias computacionais de apoio ao usuário final não ofereciam, nas etapas iniciais de projeto, ferramentas com interação adequada para as atividades mais importantes destas etapas, especialmente o desenho à mão livre (JUCHMES, LECLERCQ e AZAR, 2005). Os softwares CAD, de modelagem 3D e simulação de desempenho mais usados em projetos de arquitetura e urbanismo continuam sendo utilizados em computadores desktop e laptop (FRANCO, 2014), baseados na interação quase exclusivamente com mouse e teclado. Esta interação é rígida por uma série de limitações, começando por uma, inerente ao design do mouse, sua *affordance*.

Para Gibson (1967 *apud* STILL e DARK, 2013), a percepção de uma ação que pode ser executada intuitivamente com determinado objeto define-se como a *affordance* deste objeto. No uso de computadores, *affordance* é a propriedade de interação entre o usuário e o dispositivo, baseada na percepção que o usuário tem das características do dispositivo (STILL e DARK, 2013). O mouse possui *affordance* para apontar ou clicar botões, como as ações requeridas para a elaboração de desenhos técnicos em softwares CAD (BHAVNANI e JOHN, 2001). Pela posição que adota a mão acima do mouse, e os movimentos possíveis do mouse em relação com os movimentos percebidos pelo usuário na tela do computador, este dispositivo não oferece uma boa relação de percepção de controle no livre arrasto do ponteiro. O arrasto é a ação requerida para criar os traços que compõem o sketch (BEAUDOUIN-LAFON, 2004). Em outras palavras: o mouse é inapropriado para elaborar o tipo de representação necessário nas etapas iniciais de projeto, ou seja, inapropriado para o desenho de sketches (BUXTON, 1986).

Estando o mouse localizado fora do campo visual onde se estabelece a relação cognição-representação, o desenhista se vê obrigado a fazer um esforço extra de sincronização. É uma sincronização entre o movimento da mão no mouse e o correspondente movimento do ponteiro na tela do computador, como dois agentes distintos, respondendo a regras diferentes, atuando numa mesma atividade (BHAVNANI e JOHN, 2001). No ciclo desenhar, inspecionar, revisar, para cada início de uma iteração de desenho são necessários ao menos dois gestos físicos: o usuário movimenta o mouse, inspeciona o conseqüente movimento do ponteiro, provendo de um feedback do primeiro movimento, que informa ajustes de motricidade fina num segundo movimento do mouse, com o qual é iniciada a editoração. Esta interposição limita a continuidade entre cognição e representação, que é chave para o avanço do projeto em iterações do ciclo desenhar, inspecionar, revisar (KALLIO, 2005).

A interação humano-computador, que acontece nestas atividades instrumentada por mouse e teclado, faz parte do paradigma de computação desktop (DIX, FINLAY *et al.*, 2003), que está sendo ultrapassado pela computação móvel e a ubíqua (HEIM, 2008). Os novos equipamentos e softwares, com diferentes modos de interação instrumentados por dispositivos, entre os quais temos as canetas *stylus* e as telas *multi-touch*, fazem parte de uma tendência de aproximação de hardwares das mídias utilizadas por arquitetos e designers nas fases iniciais de projeto (FRANCO, 2014).

Company, Contero *et al.* (2006) demonstraram as qualidades da interação por caneta, com um estudo comparativo entre as mídias e ferramentas acima citadas e as mídias tradicionais, envolvendo designers iniciantes e experientes. O estudo valorou a eficiência,

segundo o tempo de elaboração de sketches; e a usabilidade, a partir da resposta qualitativa dos usuários. O trabalho de desenho de sketches com lápis e papel se registrou como “mais intuitivo e mais rápido que o CAD”, especialmente em projetistas experientes. Entre as principais avaliações, também registrou-se o fato de “permitir improvisações e imperfeições e ser confortável na mão”. O mesmo trabalho desempenhado com caneta *stylus* no computador *tablet*, se registrou como vantajoso pela sua similaridade com o lápis tradicional, com o qual comparte a mesma *affordance* para escrever e desenhar, além de ser “rápido, limpo e fácil de apagar” (*ibid.*).

A adoção de computadores *tablet*, além de fomentar a utilização de canetas *stylus*, contribui com a familiarização dos gestos *multi-touch*, geralmente utilizados em aplicativos de texto e imagens, mas com princípios de utilização comuns também para tarefas associadas ao desenho projetual, principalmente em temas de navegação (ex.: zoom e *pan*) e de edição (ex.: mover, rotacionar escalar).

Hinckley, Yatani *et al.* (2010a) (2010b) abordam o problema de desenvolvimento de software de anotações gráficas (diagramas e *storyboards*) que, explorando estas tecnologias, oferecem um novo modo de interação bimanual (com ambas mãos), combinando simultaneamente caneta e *multi-touch*. Este modo misto, ou interação multimodal, foi inspirado nas atividades criativas do trabalho bimanual nas mídias tradicionais, que combinam várias ferramentas como lápis, régua e tesouras e ações de manipulação de folhas de papel. Este trabalho demonstrou que, para os usuários, a facilidade de uso de caneta e *multi-touch* se percebe igual à das mídias e ferramentas tradicionais (HINCKLEY, YATANI *et al.*, 2010b). Isto pode trazer um grande ganho, considerando que esta implementação experimental consegue a naturalidade de uso das mídias tradicionais com os benefícios do ambiente digital, tais como: armazenamento e portabilidade da informação, facilidade de comunicação, reprodução e edição em outras oportunidades e, mais importante, a possibilidade de transição direta entre as etapas iniciais e as etapas avançadas de projeto (desde que suporte os formatos de arquivo apropriados, tipicamente formatos CAD).

Affordance a partir das simulações de desempenho

Nas tomadas de decisões feitas nas diferentes etapas de projeto, o projetista interage com o desenho na progressiva construção mental da ideia de projeto. Nesta interação, o projetista aplica conhecimentos implícitos para melhorar e modificar o desenho, significando avanços e alterações de projeto (OXMAN, 1997). Conhecimentos relacionados ao desempenho da edificação são aplicados nas tomadas de decisões, e externalizadas nos desenhos (WOO,

CLAYTON *et al.*, 2004). Quando o projetista não conta com softwares de simulação de desempenho que explicitem essa aplicação de conhecimentos, a relação se mantém implícita, porém, subjetiva. Ao utilizar softwares de simulação de desempenho, o projetista aproveita a sua inteligência visual para relacionar o conhecimento implícito com os valores de desempenho representados na simulação para tomar decisões melhor informadas por dados objetivos (OXMAN, 2009).

Quando os resultados de desempenho forem desfavoráveis, é necessária uma reformulação do projeto. Na visualização das características da simulação como consequências das decisões fundamentais de projeto (como por exemplo, as dimensões da volumetria do partido arquitetônico), o projetista torna estas características elementos de *feedback* para reformular estas decisões, que eventualmente atualizam os dados da análise, redefinindo a simulação (CRAWLEY, HAND *et al.*, 2008). Assim, podemos entender esta relação entre a simulação e o projeto, através da interação cíclica entre o projetista e o modelo simulado, baseada na percepção que o projetista tem das características da simulação (OXMAN, 2009). A partir da simulação de desempenho, o modelo possui *affordance* para direcionar modificações iterativas no projeto para melhoria do seu desempenho (STILL e DARK, 2013). Este tipo de *affordance*, característica dos modelos de representação gráfica e/ou física, seria melhor aproveitado nas etapas iniciais de projeto, durante a formulação do partido arquitetônico, para guiar o controle sobre o desempenho do produto. Ao serem feitas em etapas intermediárias e finais de projeto, as revisões e modificações necessárias para melhorar o desempenho contemplam dados mais específicos do que nas etapas iniciais, requerendo mais tempo, esforço e, principalmente, mais custos devido a alterações de investimentos previamente realizados.

DISPONIBILIDADE DAS TECNOLOGIAS DE SIMULAÇÃO DE DESEMPENHO E ENTRADA POR CANETA E *MULTI-TOUCH* PARA ARQUITETOS

Dentre os softwares de simulação de desempenho disponíveis para usuários finais, o Autodesk Ecotect se especializou em análises de desempenho ambiental (VANNINI, 2011). As atividades necessárias para preparar as simulações neste tipo de softwares são demoradas e mais apropriadas em etapas avançadas de projeto de arquitetura⁷. O software Autodesk Vasari⁸, ainda

⁷ Para a elaboração das simulações, este software requer uma prévia definição dos parâmetros geográficos e climatológicos e preparação do modelo 3D por meio da importação de modelos elaborados em outros softwares, ou modelagem direta em Ecotect, com ferramentas de modelagem limitadas. Adicionalmente, é necessário definir explicitamente materiais de revestimento.

⁸ AUTODESK, INC. **Autodesk Vasari**. Disponível em: <<http://autodeskvasari.com/>>. Acesso em: 18 jun 2014.

em desenvolvimento (versão Beta), concebido para apoiar simulações de desempenho ambiental nas etapas iniciais de projeto de arquitetura (NOME e QUEIROZ, 2013) foi construído para interoperar com softwares de assistência em etapas avançadas de projeto e gestão de obra, usando metodologia BIM (*Building Information Modeling*). Para atingir a interoperabilidade requerida pelo BIM, o Vasari trabalha num formato que obriga o projetista a antecipar questões pertencentes a etapas avançadas como, por exemplo, atributos de construção de elementos arquitetônicos específicos (*op. cit.*). Com isto força o projetista a abandonar o foco nos problemas das etapas iniciais, truncando a fluidez do trabalho das etapas iniciais de projeto.

Infelizmente, não estão disponíveis, para usuários finais, softwares para suporte à elaboração de sketches arquitetônicos em combinação com simulações de desempenho. Os softwares disponíveis pressupõem a entrada de dados cuja natureza obstaculiza as atividades de desenho à mão livre. Como visto, estas atividades são imprescindíveis na abordagem dos problemas mal definidos das etapas iniciais de projeto, quando predomina a representação de ideias difusas, plausíveis de reinterpretação. Em contrapartida, a caneta *stylus*, que permite desenhos digitais à mão livre, e outras tecnologias de entrada direta, como telas *multi-touch*, está disponível para usuários finais. Estas tecnologias tem alto grau de adoção, especialmente desde a introdução dos *smartphones* e computadores *tablet* (HAMBLEN, 2010) (INGRAHAM, 2013). Estudos previam a aceitação destas tecnologias em computadores *tablet* (ANDERSON, SCHWAGER e KERNS, 2006) (EL-GAYAR e MORAN, 2006), antecipando a familiaridade dos usuários ao trabalho com este tipo de interação (BUXTON, 2013).

A implementação das funcionalidades de entrada direta (caneta e tela *multi-touch*) também atingiu algumas áreas de softwares para usuários finais. Desenvolvedores dos pacotes de software mais tradicionalmente representativos do trabalho com imagens digitais, como o Adobe Photoshop, já suportam entrada por caneta *stylus*, oferecendo ainda funcionalidade específica segundo os diferentes níveis de pressão imprimida na caneta (LEGASPI, 2010). Esta funcionalidade permite, por exemplo, controlar a largura do traço do pincel ou a transparência, e assim conseguir um maior controle das características da pintura num só gesto, similar às operações com tinta sobre papel (*ibid.*). Aplicativos com estas funcionalidades têm permitido que as tecnologias de caneta e tela *multi-touch* apresentem alto grau de adoção em disciplinas criativas como as artes visuais e a ilustração (SAORÍN, DE LA TORRE *et al.*, 2011), com presença ativa e organizada dos usuários destas tecnologias em redes sociais e profissionais (DICHERRI, 2013).

CARACTERIZAÇÃO DO PROBLEMA DA PESQUISA

Os aplicativos com interação por caneta e *multi-touch*, direcionados para arquitetura e urbanismo, ainda não se centram na modelagem do projeto; mas na visualização de modelos (navegação/passeio virtual, simulação de sombras etc.). Com exceção do Autodesk FormIt⁹, ainda não suportam a criação de modelos 3D (KANG, KIM *et al.*, 2013) e não apoiam a transição automática entre as etapas iniciais e as etapas avançadas de projeto de arquitetura em meio digital.

Merecem citação as aplicações tecnológicas (softwares ou estratégia de uso de softwares) voltadas para a arquitetura, com suporte às atividades das etapas iniciais de projeto¹⁰. Dentre estas, destaca-se o trabalho de Pranovich (2004) sobre o suporte para a elaboração de desenhos esquemáticos de plantas arquitetônicas segundo relações entre espaços. Juchmes, Leclercq e Azar (2005) desenvolveram um método de reconhecimento de elementos da linguagem gráfica arquitetônica em sketches em planta. Kallio (2005) desenvolveu um software de elaboração de sketches em 3D a partir do desenho sobre um plano auxiliar de livre manipulação. Yu e Zhang (2007) definiram um procedimento de geração de modelos 3D baseados no modo de projetar típico do arquiteto, a partir de plantas e cortes. Elsen e Leclercq (2008) desenvolveram um sistema de software e hardware para interação e comunicação entre membros de equipes de projeto que inclui o suporte de sketches colaborativos à distância. Por meio do reconhecimento de sketches, estes cinco exemplos abordam o auxílio ao projetista na elaboração de soluções aos problemas mal definidos das etapas iniciais de projeto, porém não abordam a transição da representação inicial do projeto para os ambientes CAD, ambientes onde se processam as etapas avançadas.

Righi e Celani (2008) estudaram a reestruturação dos processos de projeto arquitetônico apoiada no uso de dispositivos de entrada direta (incluindo caneta sobre *tablet* opaca) como instrumentos de projeto. Alaçam e Çagdas (2008) estudaram o suporte de entrada bimanual na modelagem arquitetônica, testando combinações de mouses e caneta. Estas duas experiências modificam os métodos de entrada do ambiente CAD, provendo interações mais apropriadas para as etapas iniciais de projeto. Para o projetista, trabalhar nas etapas iniciais, no formato CAD necessário nas etapas avançadas, evita a distorção e ineficiência da transição de meios de

⁹ AUTODESK, INC. **Autodesk FormIt**. Disponível em: <<http://autodeskformit.com/>>. Acesso em: 5 mai 2014.

¹⁰ Há uma diferença significativa na abordagem dos sketches iniciais segundo a área de projetos: em projetos de arquitetura se inicia principalmente com desenhos em planta ou em corte (YU e ZHANG, 2007) (ELSEN e LECLERCQ, 2008); enquanto no design de produto os sketches iniciais costumam ser em perspectiva (ELSEN e LECLERCQ, 2008). Por esta razão, no capítulo 2 faz-se uma revisão da literatura considerando principalmente os trabalhos orientados para projetos de arquitetura.

representação ao avançar nas etapas de projeto. Porém, estas experiências não abordam a elaboração de sketches, própria das etapas iniciais.

Fora da área da arquitetura, são relevantes: Masry e Lipson (2005); Naya, Contero *et al.* (2008); Lopes, Mendes *et al.* (2011); e Kang, Kim *et al.* (2013). Os quatro utilizam a modelagem baseada em sketch como técnica que oferece uma elaboração de modelos simples de maneira mais rápida e intuitiva que a possível em softwares CAD, ao mesmo tempo que oferecem compatibilidade com ambientes CAD. Excetuando Masry e Lipson (2005), os outros trabalhos previamente citados descartam o formato digital dos modelos gerados como estratégia para oferecer simulações preliminares de desempenho que potencializem a abordagem de problemas de desempenho nas etapas iniciais de projeto. Os dois primeiros (MASRY e LIPSON, 2005) (NAYA, CONTERO *et al.*, 2008) exploram a interação por caneta em computadores *tablet*; e o último (KANG, KIM *et al.*, 2013), a interação *multi-touch* em dispositivos móveis. Porém, não suportam simultaneamente os dois modos de entrada. Somente em Lopes, Mendes *et al.* (2011) registra-se a exploração dos protocolos bimanuais multimodais que Hinckley, Yatani *et al.* (2010a) (2010b) implementaram para anotações gráficas, para o aproveitamento das tecnologias *multi-touch* na modelagem 3D a partir de sketches.

Em softwares com interação baseada em gestos *multi-touch*, a implementação de protocolos multimodais melhorariam o vocabulário de gestos de edição do modelo, especialmente nas funcionalidades de transformação (mover, rotacionar, escalar, deformar, cortar, entre outras), conseguindo adicionar estas funcionalidades sem aumentar a complexidade da interação (*ibid.*). A utilização de softwares que incluíssem simulações preliminares de desempenho nas etapas iniciais de projeto arquitetônico possibilitaria a definição de partidos arquitetônicos melhor fundamentados do que aqueles baseados somente em conhecimento implícito (OXMAN, 2009). Em comparação com os softwares atualmente disponíveis, uma aplicação de software que aproveite estas tecnologias, não só permitiria os benefícios informacionais, documentais e comunicativos do trabalho no ambiente digital; principalmente, aumentaria as possibilidades do projetista de antecipar a solução de problemas de projeto.

Ainda que não abordem a integração destas funcionalidades, os trabalhos estudados trazem aportes (listados a seguir) que devem ser considerados no desenvolvimento de critérios de aproveitamento destas tecnologias, como a modelagem baseada em sketch. Resumindo, a literatura descreve as seguintes funcionalidades:

- Desenho de sketches digitais arquitetônicos em 2D e 3D (KALLIO, 2005) (YU e ZHANG, 2007);

- Reconhecimento de elementos gráficos (símbolos) e interpretação no contexto arquitetônico (PRANOVICH, 2004) (JUCHMES, LECLERCQ e AZAR, 2005) (YU e ZHANG, 2007);
- Usabilidade do *stylus*/tablet no desenho de sketches digitais (NAYA, CONTERO *et al.*, 2008) (KANG, KIM *et al.*, 2013);
- Interação multimodal caneta+*touch* (HINCKLEY, YATANI *et al.*, 2010a) (2010b) (LOPES, MENDES *et al.*, 2011);
- Manipulação *multi-touch* de modelos 3D (LOPES, MENDES *et al.*, 2011) (KANG, KIM *et al.*, 2013);
- Simulação de desempenho estrutural em modelos 3D baseados em sketches (MASRY e LIPSON, 2005);
- Interoperabilidade do sketch com formatos CAD (NAYA, CONTERO *et al.*, 2008) (KANG, KIM *et al.*, 2013).

A abordagem, por parte do projetista, dos problemas mal definidos de projeto, a emergência de formas e a re-representação são possíveis com o emprego de funcionalidades para a elaboração de modelos em 3D a partir de sketches visualmente ambíguos (KALLIO, 2005) (YU e ZHANG, 2007) e desenhados à mão livre com caneta *stylus* (NAYA, CONTERO *et al.*, 2008). O aproveitamento da linguagem gráfica do arquiteto é possível por meio do reconhecimento de formas geométricas e símbolos desenhados à mão livre (PRANOVICH, 2004) (JUCHMES, LECLERCQ e AZAR, 2005) (YU e ZHANG, 2007). O desenvolvimento do modelo 3D pode ser automatizado a partir do desenho à mão livre por meio de funcionalidades de modelagem baseada em sketch, cuja manipulação e edição podem ser controladas por gestos multimodais que aproveitam a entrada simultânea de caneta e *multi-touch* (LOPES, MENDES *et al.*, 2011). Finalmente, os projetistas podem abordar decisões de projeto direcionadas ao bom desempenho desde as etapas iniciais, por meio de simulações atuando sobre modelos construídos a partir de sketches (MASRY e LIPSON, 2005), gerados num formato compatível com o ambiente CAD onde continuará a elaboração do projeto nas etapas avançadas (NAYA, CONTERO *et al.*, 2008) (KANG, KIM *et al.*, 2013).

As funcionalidades acima descritas foram desenvolvidas de forma isolada entre si e, dos oito trabalhos citados, três – Naya, Contero *et al.* (2008); Lopes, Mendes *et al.* (2011) e Kang, Kim *et al.* (2013) – não estão voltados para a arquitetura. Os softwares apresentados por Naya, Contero *et al.* (2008) e Kang, Kim *et al.* (2013), abordam a modelagem de peças de engenharia mecânica através de traços simples e desenhos gestuais que adaptam funcionalidades de modelagem de softwares MCAD (desenho mecânico assistido por computador). As funcionalidades apresentadas por Lopes, Mendes *et al.* (2011), adaptam a modelagem de sólidos de forma livre, típicos do design de personagens, para o seu uso multimodal de caneta e *multi-touch*. Portanto, arquitetos não contam com ambiente computacional que ofereça, de

forma integrada, o conjunto destas funcionalidades específicas para apoiar o processo de projeto de arquitetura. Sem o apoio digital às atividades das etapas iniciais, as ferramentas e suportes tradicionais, como lápis e papel, continuarão sendo preferidas pelos arquitetos em relação às tecnologias digitais. A falta de aproveitamento das aplicações tecnológicas, não vai ao encontro da evolução dos processos projetuais que exigem desempenho cada vez mais eficiente, num prazo cada vez menor. Assim, o objetivo deste estudo é definir requisitos de um ambiente digital que aumente o apoio que as tecnologias CAD oferecem ao arquiteto nas etapas iniciais de projeto. Para atingir este objetivo se faz descrever, de forma estruturada:

- O estado da arte da aplicação de tecnologias de software e hardware para as etapas iniciais de projetos de arquitetura.
- Critérios de interação simultânea de caneta e gestos *multi-touch* específicos para as atividades das fases iniciais de projetos de arquitetura e urbanismo.
- Um procedimento que permita a geração de modelos 3D a partir de sketches arquitetônicos, e a transição automática entre as etapas iniciais e as etapas avançadas de projeto.

Esta dissertação busca, a partir das descrições acima, explicitar os requisitos e potencialidades de um ambiente que integre diferentes funcionalidades de apoio às etapas iniciais de projeto de arquitetura e está dividida em quatro partes. A primeira parte (1) discorre sobre os fundamentos do problema formulado, ordenados em três eixos: técnicas e ferramentas de projeto, interação humano-computador relevante às atividades projetuais, e adoção de tecnologias de caneta e *multi-touch*. A segunda parte (2) envolve a análise comparativa entre os softwares experimentais que conformam o estado da arte da pesquisa no tema. A terceira parte (3) descreve as funcionalidades de software que abordam o aproveitamento da tecnologia de caneta e tela *multi-touch* numa interação apropriada às etapas iniciais de projeto de arquitetura. Na quarta e última parte (4), são traçadas conclusões sobre a aplicação destas funcionalidades e propostos trabalhos futuros, visando o desenvolvimento do conhecimento sobre estas tecnologias.

1 FUNDAMENTAÇÃO TEÓRICA

1.1 TÉCNICAS E FERRAMENTAS RELEVANTES EM PROJETO

Os projetistas acostumam classificar técnicas de desenho segundo a ferramenta de software utilizada. A classificação destas técnicas e das ferramentas associadas segue diferentes denominações segundo a etapa e área de projetos, mas guardam a mesma relação de progressão: desde a etapa mais difusa até a etapa mais precisa. Como foi mencionado na introdução, no projeto arquitetônico destacam-se as etapas de partido arquitetônico, estudo preliminar, anteprojeto e projeto executivo (MOREIRA e KOWALTOWSKI, 2009). O partido arquitetônico e o estudo preliminar são as etapas iniciais, onde se constitui a configuração inicial da solução arquitetônica, considerando as principais exigências contidas na lista de requerimentos, denominada **programa de necessidades** (*ibid.*).

1.1.1 Técnicas de desenho mais comuns nas etapas iniciais de projeto

No processo de projeto, o projetista, começa tipicamente com uma recompilação de fatores contextuais (ex.: clima, topografia, entorno urbano, limitações e imposições da legislação, requerimentos do usuário), identificação das necessidades espaciais e listagem do programa de necessidades, que o levam à formulação dos problemas de projeto. Com estas informações, inicia o esboço de ideias com a técnica de desenho à mão livre (YU e ZHANG, 2007).

As ferramentas de suporte tradicionais, lápis e papel, têm servido para definir e comunicar ideias criativas através do desenho à mão na arquitetura desde o Renascimento (CARPO, 2013). Hoje em dia, após décadas da introdução do computador nos processos de projeto, lápis e papel continuam a ser os mais utilizados na elaboração de sketches durante as etapas iniciais de projeto (FLORIO, 2010).

O lápis, como ferramenta para representação de projetos, possui *affordances* para a escritura, o desenho de sketches e o desenho técnico¹¹. A morfologia do lápis, simples como é, se adapta à anatomia da mão e, com a prática no seu uso desde a pré-escola, o projetista o percebe como um instrumento natural de trabalho. Para elaboração de desenhos, existem diferentes lápis: lápis com grafites duros, de graduações da série H (H, 2H... 9H), com os quais se elaboram traços finos e claros; e lápis com grafites macios, de graduações da série B (2B...

¹¹ O desenho técnico a lápis, que emprega instrumentos como régua e esquadros em prancheta, é ainda vigente no ensino da arquitetura, engenharia e design, e em alguns escritórios profissionais mais tradicionais (TAMASHIRO, 2003).

6B), com os quais se consegue traços grossos e visualmente pesados (YEE, 2009). Grafites duros são apropriados para traçar linhas auxiliares ou linhas de construção do desenho e os grafites macios são apropriados para o desenho de linhas definitivas ou traços contrastantes acima de linhas auxiliares (TAMASHIRO, 2003). Grafites com dureza média-macia (graduação B, HB ou F), oferecem versatilidade, ao exibir comportamentos distintos quando utilizados com diferentes intensidades de pressão sobre o papel: o traço realizado com uma pressão leve, exibe uma aparência similar à feita com um lápis da série H, e o traço com uma maior pressão permite a expressão tipicamente pesada dos lápis da série B. Esta propriedade faz, dos lápis de dureza média-macia, instrumentos úteis no desenho de sketches, ao permitir traços distintos segundo a pressão imprimida no papel, sem ter que deter o processo para trocar de lápis (ou trocar de grafite na lapiseira). Assim, o projetista pode demarcar formas emergentes no instante em que as identifica e, em termos gerais, mudar sobre a marcha o nível de precisão com o qual aborda os problemas de projeto, seja dentro de uma iteração de desenho, seja através da sequência de representações¹².

Após algumas iterações de desenho à mão livre, a sequência de sketches produzida, conta com informação suficiente para definir o encerramento de uma etapa de projeto e o começo da seguinte (COMPANY, CONTERO *et al.*, 2006), na qual o projetista se aproxima de problemas mais específicos que, até então, não eram suficientemente definidos (OXMAN, 2002).

Considerado o uso mais representativo das técnicas de representação de projeto, a etapa de concepção tem lugar no ambiente digital, empregando a técnica de modelagem em 3D (DENZER e GARDZELEWSKI, 2011). Com esta técnica se produz um modelo visualizado na tela do computador como um conjunto de sólidos ou superfícies, internamente descrito pelas coordenadas cartesianas dos vértices destas entidades geométricas (também denominado modelo vetorial) (MONEDERO, 2000). Este nível de definição necessita de precisão numérica relacionada com tomadas de decisão mais específicas que as do partido ou design conceitual. Devido a complexidade geométrica (ex.: objetos com características definidas em mais de dois planos de referência), ou devido a existência de compartilhamento de tarefas por desenhistas (ex.: quando o modelo 3D vai ser elaborado por outra pessoa da equipe, que deva interpretar informações difusas), para abordar o nível de definição do modelo 3D, a modelagem frequentemente envolve prévia elaboração de desenhos CAD auxiliares em 2D (COMPANY, CONTERO *et al.*, 2006). Como ilustrado na Tabela 1, em arquitetura, a modelagem em 3D é

¹² Veja-se na página 5 da Introdução.

hoje predominantemente feita no software Trimble SketchUp¹³, e o desenho CAD, utilizando tipicamente o software Autodesk AutoCAD (WINSTON, 2010).

Tabela 1: Técnicas, ferramentas e suportes comumente associados a tipos de representação em projetos de arquitetura.

Tipo de representação	Técnica de elaboração	Ferramenta e suporte representativos
Sketch	Desenho à mão livre	Lápis sobre papel
Modelo 3D	Modelagem em 3D	SketchUp em computadores desktop
Simulação de desempenho	Técnicas de simulação de desempenho	ANSYS; Ecotect em computadores desktop
Desenho técnico	Desenho CAD	AutoCAD em computadores desktop

Fonte: o autor, baseado em Crawley, Hand *et al.* (2008); Murugappan e Ramani (2009); e Winston (2010).

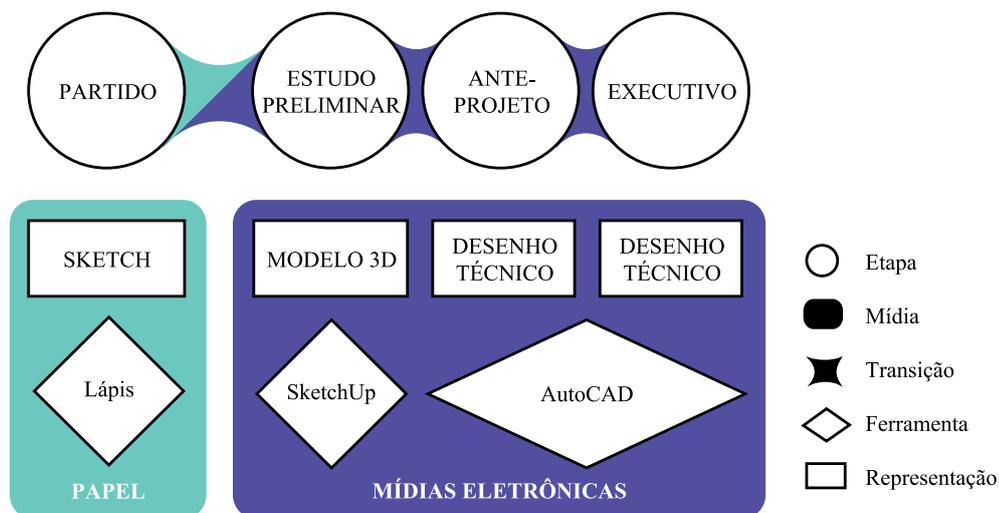
Neste fluxo de trabalho representativo, para poder continuar evoluindo o projeto com desenhos digitais, o projetista deve primeiro fazer a transferência da informação da mídia analógica para a digital (YU e ZHANG, 2007). Como adiantado na introdução¹⁴, estes procedimentos de transferência de desenhos, o de vetorização por redesenho e ainda o de reconstrução indireta, implicam um esforço adicional de interpretação por parte do desenhista (KELLY e GERO, 2009). A perda de informação e o atraso, consequências deste esforço adicional, tornam o processo de representação ineficiente (COMPANY, CONTERO *et al.*, 2006).

Na transferência entre as mídias física e digital (Figura 4), ocorre a descontinuidade do meio em que os desenhos produzidos, aumentam as tarefas interpretativas (MITCHELL e MCCULLOUGH, 1995), tornando o processo demorado e ineficiente dependendo de quão ambíguos ou difusos são os sketches iniciais.

¹³ SketchUp, software que pertencia à Google no momento da pesquisa de Winston (2010), foi posteriormente adquirido pela Trimble Navigation Ltd., assim como a sua equipe de desenvolvimento de software (TRIMBLE NAVIGATION LTD., 2013).

¹⁴ Veja-se a descrição do problema da transferência entre diferentes mídias, na Introdução, página 8.

Figura 4: Transição de mídias no processo de projeto de arquitetura.



Fonte: o autor.

1.1.2 Tipos de sketch segundo a atividade projetual

Ferguson (1992 *apud* COMPANY, CONTERO *et al.*, 2006), considera uma classificação dos tipos de sketches, em que, segundo a atividade projetual, eles podem ser sketches de pensamento, sketches comunicativos e sketches prescritivos.

Sketches de pensamento são elaborados pelo projetista para focalizar e guiar o seu pensamento não verbal. O seu conteúdo varia entre informações diagramáticas, definições formais e representacionais, mas não estão regidas por convenções e seguem uma lógica pessoal de cada projetista. Geralmente são difíceis de interpretar por outras pessoas além do autor do sketch, que os faz como estruturas mnemônicas para ajudar a sua inteligência visual a tomar decisões e gerar ideias que possam ser encaminhadas em soluções de projeto (SUWA, PURCELL e GERO, 1998). Com este tipo de sketches se abordam as ideias iniciais de volumetria, relações entre volumes e relações topológicas (ex.: quantidade de lados, arestas, furos, relação destas características topológicas entre elementos e delimitação de espaços). Pesquisas para a interpretação por computador deste tipo de sketch se apoiam em complexos métodos de reconhecimento de padrões e lógica difusa (GERO e YAN, 1993) (FONSECA e JORGE, 2001).

Sketches comunicativos são empregados para suportar a discussão entre as pessoas envolvidas no projeto (projetistas, clientes, assessores ou professores, etc.). Este tipo de sketch é elaborado no meio de uma conversa sobre o projeto. É um material de apoio na comunicação de ideias de projeto, e sua interpretação é principalmente clara para as pessoas envolvidas na conversa, mas não necessariamente interpretáveis por terceiros pessoas. Geralmente, estes sketches se baseiam em definições formais e representacionais do objeto projetado e podem incluir informações gráficas e anotações gestuais e textuais que enfatizem o discurso do

projetista. Modelos de interpretação destes sketches incluem métodos de reconhecimento de texto e gestos gráficos (DO, 1999) e métodos de interpretação baseada em conhecimento (FORBUS, USHER *et al.*, 2008), mas em geral, não são muito diferentes dos métodos assinalados para a interpretação de sketches de pensamento (COMPANY, CONTERO *et al.*, 2006).

Sketches prescritivos são elaborados principalmente em equipes de projeto, para instruir os desenhistas de CAD sobre como fazer modelos em 3D ou desenhos técnicos. Mesmo sendo elaborados à mão livre, estes desenhos costumam ser documentados com linhas de cota e outros elementos convencionais do desenho técnico. Eles são muito importantes no processo de projeto. Este tipo de desenho costuma ser elaborado pelos chefes de projeto e, ao serem passados aos desenhistas CAD, constituem um instrumento de transição das etapas iniciais de projeto às avançadas (COMPANY, CONTERO *et al.*, 2006). Devido a sua função informativa, este tipo de desenho é menos difuso, diminuindo a dificuldade de interpretação (*ibid.*). Por esta razão, têm sido usados na literatura para tentar incorporar este processo nas mídias digitais (COMPANY, CONTERO *et al.*, 2003) (JUCHMES, LECLERCQ e AZAR, 2005) (COMPANY, PIQUER e CONTERO, 2004) (YU e ZHANG, 2007). O sucesso na incorporação do desenho de sketches não está só determinado pelo desenvolvimento de procedimentos de modelagem baseada em sketch, mas também pela instrumentação de hardware que permite a interação apropriada. Para poder abordar este tema, é necessário entender o paradigma de interação humano-computador no qual estas experiências se baseiam.

1.2 INTERAÇÃO ENTRE O PROJETISTA E O COMPUTADOR

Como foi previamente identificado, estes projetistas, usuários de computadores, na sua grande maioria, empregam ferramentas de CAD (DENZER e GARDZELEWSKI, 2011). A interação com estas ferramentas é instrumentada por mouse e teclado como dispositivos de entrada. Estes dispositivos de entrada funcionam com base num paradigma de interação que está sendo ultrapassado por outros (BEAUDOUIN-LAFON, 2004), que serão explicados na sequência. Isto é especialmente relevante no que respeita às funções que estas ferramentas cumprem nas etapas iniciais de projeto (KANG, KIM *et al.*, 2013). Esta situação tem uma causa relacionada com os efeitos históricos das mudanças de paradigma (BUXTON, 2008).

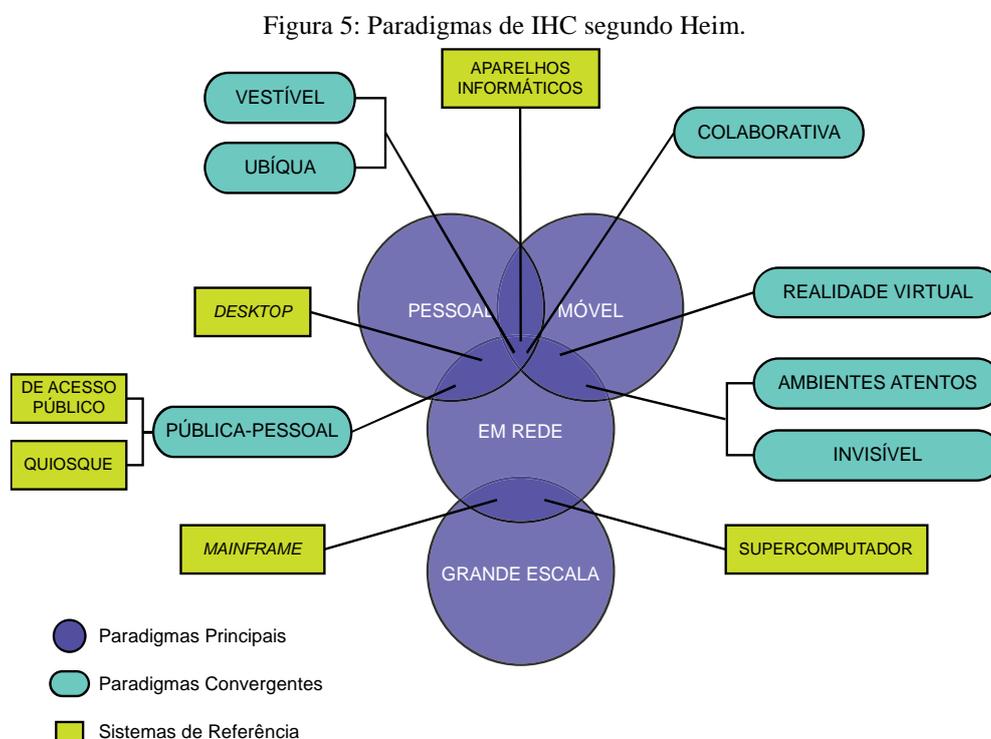
1.2.1 Paradigmas de interação humano-computador

A maneira como interagimos com o computador tem mudado ao longo da história da computação. Esta evolução é marcada por avanços pontuais significativos, ou mudanças de

paradigma. Na literatura, existem diferentes classificações de paradigmas que, devido à natureza interdisciplinar desta pesquisa, consideram-se relacionadas. Nas ciências da computação, se estudam, entre outros, os paradigmas de interação humano-computador (IHC). Nesta área de conhecimento, um paradigma de IHC é um “modelo ou padrão de interação humano-computador que engloba todos os aspectos de interação, incluindo físicos, virtuais, perceptivos e cognitivos” (HEIM, 2008).

Como se mostra na Figura 5, para Heim (2008), estes paradigmas são principalmente: computação em grande escala, computação em rede, computação pessoal e computação móvel. Na convergência destes paradigmas principais, existem: computação público-pessoal; ambientes colaborativos; realidade virtual; ambientes atentos e computação invisível (ou transparente); e computação ubíqua e vestível (*op. cit.*).

Na documentação desta classificação costuma-se dar exemplos de sistemas significativos para determinado paradigma. O computador *desktop* é a primeira referência de tipo de sistema que exemplifica a confluência de paradigmas de computação pessoal e computação em rede ao levar as funções dos antigos computadores *mainframe*, incluindo as funções em rede, ao espaço de interação do terminal de computador (HEIM, 2008).



Fonte: Heim (2008) (tradução nossa).

Outro exemplo relevante para esta pesquisa é o sistema de aparelhos informáticos, como os *smartphones* e os computadores *tablet*, na confluência entre computação pessoal e móvel (*op. cit.*).

Dix, Finlay *et al.* (2003) apresentam uma classificação mais extensa, na qual não fazem distinção entre paradigmas e sistemas de referência¹⁵, mas os ordenam cronologicamente, destacando a sua importância histórica enquanto mudanças de paradigma. Entre os mais relevantes podemos nomear (e posteriormente aprofundar): processamento em lote e compartilhamento de tempo de computação (*time-sharing*, começo da computação interativa), visualização gráfica (ou manipulação gráfica), computação pessoal, interface WIMP (*windows, icons, menu and pointer*), manipulação direta, multimodalidade, e computação ubíqua (*ibid.*).

Considerando a sobreposição das classificações de paradigmas entre Dix, Finlay *et al.* (2003) e Heim (2008), a Figura 6 apresenta um recorte destes paradigmas, combinando-os num mapeamento ordenado cronologicamente.

Figura 6: Mapeamento dos paradigmas de IHC relevantes.



Fonte: o autor, baseado em Dix, Finlay *et al.* (2003) e Heim (2008).

Na década de 1950, o desenvolvimento e instalação de grandes computadores em algumas empresas (como a General Electric) e instituições de pesquisa (como o MIT) permitiu o processamento em lotes de grandes quantidades de informação, através de uma interação de “linha de comandos” (DIX, FINLAY *et al.*, 2003). Neste modo de interação, o programador introduzia comandos textuais num terminal do computador, o qual se compunha de um teclado e uma pequena tela monocromática de tubo de descarga (CTR) que apresentava a informação textual (HEIM, 2008).

¹⁵ Os autores (DIX, FINLAY *et al.*, 2003) comentam que nem todos os sistemas listados por eles representam mudanças de paradigma. Desde a ótica de Heim (2008), muitos dos que ele considera sistemas de referência são, para Dix *et al.* (2003), o único expoente de um paradigma, podendo ser usado para nomear o paradigma em si.

Com a invenção das redes de computador, foi possível trabalhar com mais de um terminal, que não precisava estar localizado no mesmo recinto que o computador central ou *mainframe*, abrindo a possibilidade de uso a mais pessoas (MITCHELL e MCCULLOUGH, 1995). Para permitir esta possibilidade, foi desenvolvido o time-sharing, sistema de compartilhamento de computação do mainframe segundo um protocolo de transferência de dados bidirecional, entre o mainframe e os terminais, um por vez, segundo uma gestão de tempo de uso configurável (DIX, FINLAY *et al.*, 2003).

A implementação do time-sharing é considerada por estudiosos (*ibid.*) como o início da computação interativa.

1.2.2 Evolução da interação por teclado e mouse

O paradigma da computação pessoal, como ilustrado anteriormente, começou com a incorporação das funções do *mainframe* no computador desktop. Logo, a aparição do mouse e dos sistemas operacionais com interface gráfica: Xerox Star (1983), Macintosh (1984) e Windows (1984), permitiram a interação com elementos WIMP (*windows, icons, menus and pointer*), artefatos virtuais ainda vigentes nas interfaces gráficas para funcionar nos sistemas operacionais dos computadores desktop. A interação do mouse se baseia na manipulação gráfica destes elementos com os quais se comanda a execução das atividades na tela, através do controle do ponteiro, por meio do mapeamento do movimento relativo do mouse (BUXTON, 2013).

Em geral, na evolução dos softwares de criação e editoração de conteúdo (ex.: processadores de palavras, aplicativos de desenho, editores de conteúdo interativo), destacam-se três posturas de aproximação à interação com estes softwares: WISIWIT (*what I see is what I type*), WYSIWYG (*what you see is what you get*) e WYSIWYM (*what you see is what you mean*). No princípio WISIWIT (também conhecido como YAFIYGI - *you asked for it, you got it*), a interação é baseada em texto: o usuário digita ou seleciona um comando predefinido e o computador o executa, retornando uma resposta (RAYMOND, 1995). É um princípio herdado da interação com os terminais de computador através de uma interface de linha de comandos, recebidos pelo computador *mainframe* como mensagens imperativas. Na aproximação WYSIWYG, o usuário deve usar os recursos da interface de uma maneira mais intuitiva que os comandos textuais, apelando à denotação icônica da interface para controlar, tipicamente através do mouse, a aparência do conteúdo em editoração, que é a representação mais literal possível na mídia utilizada (REBELO, 2009). A aproximação WYSIWYM adiciona-se à anterior, mas o usuário deve combinar a utilização de recursos icônicos com ações num nível

de abstração mais aprofundado, ao utilizar recursos simbólicos (metadados ou atributos de objetos) para definir a semântica do conteúdo em editoração (JOHANSSON, 2006).

A indústria de softwares CAD tem sido grandemente influenciada pela evolução anteriormente descrita dos sistemas de interação dos computadores pessoais. O primeiro sistema comercial de software CAD, o AMD CATIA¹⁶, já contava com funcionalidades avançadas de desenho paramétrico (CARLSON, 2003). Mas a sua complexidade operacional de interação WISIWIT o limitava a computadores especializados (estações de trabalho UNIX, equipados com dispositivos de entrada como mesas digitalizadoras, similares às atuais *tablets* opacas), aos quais a população representativa de projetistas não tinha acesso. O sucesso na adoção dos sistemas operacionais com interfaces gráficas WIMP na década de 1980 facilitou a implementação de sistemas de CAD mais acessíveis (*ibid.*).

Autodesk AutoCAD se destacou como o primeiro software CAD para computadores pessoais, simplificado quanto ao processamento dos dados e quanto aos modos de interação, suportando grande parte da funcionalidade do CATIA só com teclado e mouse (COHN, 2010). Desta maneira permitiu a sua adoção massiva, definindo um padrão de trabalho nas indústrias de arquitetura, engenharia e construção (*ibid.*).

O conhecimento técnico sobre métodos de cálculo e representação destes profissionais foi transferido para os softwares CAD com uma interação de aproximação WYSIWYG, que passaram a ser usados adaptando as técnicas típicas das pranchetas de desenho (BÜRDEK, 2005). Esta adaptação de métodos não foi suficientemente direta: o traço do lápis foi substituído por combinações de movimentos e cliques do mouse. Este dispositivo é apropriado para apontar e selecionar, mas não para o arrasto do ponteiro, ação requerida para criar os traços que compõem o sketch (BUXTON, 1986). O que começou como um sucesso de adaptação às limitações de hardware acabou estagnando a evolução das metodologias digitais de projeto (BHAVNANI e JOHN, 2001).

Os primeiros estudos sobre a usabilidade dos softwares CAD já demonstravam grandes problemas de desempenho dos projetistas (BHAVNANI, GARRETT e SHAW, 1993). A crescente quantidade de funções, mesmo motivada por demandas recolhidas dos usuários, não responde adequadamente às necessidades deles (*ibid.*). Esta tendência acabou gerando uma série de funcionalidades improváveis de serem aproveitadas por projetistas em tarefas

¹⁶ No lançamento da primeira versão em 1977, a empresa desenvolvedora do CATIA se denominava Avions Marcel Dassault (AMD). Hoje em dia, a mesma equipe de desenvolvimento é Dassault Systèmes (DASSAULT SYSTÈMES). **3D CAD design software CATIA**. Disponível em: <<http://www.3ds.com/products-services/catia/>> Acesso em: 21 abr 2014).

importantes. Este fenômeno, conhecido como sobrecarga de funções¹⁷ (BÜRDEK, 2005), teve como consequência (a) a diminuição da eficiência do computador (pelo intenso uso da memória) e (b) a diminuição da usabilidade, por apresentar interfaces **cifradas** (*ibid.*).

Uma interface se torna cifrada quando, pela saturação de recursos visuais de ativação de comandos (botões, *checklists*, *sliders*, etc.), está obrigada a subdividir a funcionalidade em mais recursos de interface, muitos deles não visíveis, porém cifrados por baixo de mais níveis de **protocolos de uso** (*op. cit.*). Protocolos de uso são as sequências de ações necessárias para executar uma tarefa, que variam segundo o sistema de interação (ou software). O principal problema das interfaces cifradas é o aumento, no usuário, da **carga cognitiva** na chamada do comando que executa uma tarefa, especialmente quando a tarefa demanda esforço cognitivo (STILL e DARK, 2013), como é o caso das tarefas de projeto. Neste contexto, carga de memória (ou carga cognitiva) se refere ao esforço mental que o usuário deve fazer para lembrar-se das ações para a execução de uma tarefa, incluindo os protocolos de uso: navegar por uma interface para acionar um comando e seguir os passos de execução da tarefa. Por exemplo, em AutoCAD, para marcar pontos de divisão numa parede em partes iguais, e assim definir eixos de esquadrias de janela, é necessário interromper o raciocínio sobre elementos arquitetônicos e abstraí-lo, focando no protocolo ilustrado na Figura 7.

Figura 7: Protocolo para marcar pontos de divisão numa linha em Autodesk AutoCAD.

1	Clicar no menu Draw;
2	Clicar no submenu Point;
3	Selecionar o comando Divide;
4	O software pede ao usuário selecionar a linha a ser dividida através de uma mensagem na linha de comandos;
5	Ao selecionar a linha, uma mensagem pergunta pela quantidade de divisões;
6	Digitar o número de divisões, pressionar Enter.

Fonte: o autor.

Apesar do nome, o resultado do comando Divide não é uma divisão da linha em segmentos de linhas, e sim o desenho de uma sequência de pontos ao longo de uma linha, que só são visíveis se o usuário tiver tomado a precaução de mudar o símbolo de representação de ponto nas opções do AutoCAD (do contrário, a representação do ponto por omissão, ou por

¹⁷ Do termo em inglês: *function overload*, trazido ao discurso do design por Bürdek (2005), citando a Fischer (A Phenomenology of Electronic ‘Devices’, 2001 *apud* Bürdek, 2005), que o define como um fenômeno que acontece em produtos equipados com microprocessadores. Mesmo usado em computação, este termo não tem relação com *function overloading*, o método de programação para atribuir diferentes tarefas a uma só função (MEYER, B. Overloading vs Object Technology. **Journal of Object-Oriented Programming**, v. 14, n. 4, 2001).

default é só um pixel na tela, o que é imperceptível acima de uma linha de um pixel de espessura). Posteriormente, o usuário deve executar outros passos, como o desenho de linhas perpendiculares à linha de base nos pontos de divisão para poder visualizar um desenho que represente os eixos das esquadrias que imaginou um minuto atrás.

Bhavnani e John (2001) demonstraram que a interação permitida pelos softwares CAD nas atividades de projeto é ineficiente e precisa ser repensada. Como alternativa provisional, propuseram estratégias de uso com as quais é possível um desempenho mais rápido, mas isto só é possível adaptando os métodos em função do computador e não do usuário¹⁸ (*ibid.*).

Devido às limitações das ferramentas de desenho CAD e modelagem 3D, comparativamente ao desenho à mão livre, esta tendência é prejudicial para o projeto por uma série de razões descritas na sequência.

1.2.3 Prejuízos da utilização de ferramentas CAD nas etapas iniciais de projeto

1. **A carga adicional de memória faz o processo de projeto ineficiente:** a conexão entre a mente e o desenho CAD através do mouse faz das atividades iniciais de projeto um processo lento e ineficiente, em comparação com a conexão entre a mente e o sketch através do lápis, assim como o discutido em Company, Contero *et al.* (2006). Isto é porque o lápis permite o desenho num processo ininterrupto, mesmo que a ideia esteja mal definida. Parando o processo para definir mentalmente parâmetros de precisão CAD e seguir protocolos de uso¹⁹, o projetista aumenta a sua carga de memória (STILL e DARK, 2013) e demora mais tempo para definir uma ideia de partido (BHAVNANI, FLEMMING *et al.*, 1996) (DENZER e GARDZELEWSKI, 2011).
2. **Dificulta-se a comunicação:** quando o projetista trabalha diretamente com estas ferramentas digitais típicas, sem a elaboração prévia de desenhos a lápis, a colaboração entre projetistas e a comunicação com o cliente (ou professor) são mais difíceis, já que é quebrado o discurso visual baseado nas convenções gráficas (ASANOWICZ, 2003). Estas convenções, como os gestos visuais dos sketches comunicativos e os símbolos arquitetônicos presentes nos sketches prescritivos, servem de referência para ajudar a comunicar mais fielmente a ideia imaginada pelo projetista e evitar erros de interpretação (KELLY e GERO, 2009).

¹⁸ Por exemplo, a estratégia de “fazer organizações conhecidas pelo computador”, o usuário de CAD deve, conscientemente elaborar uma série de *layers* segundo a classe de elemento arquitetônico (BHAVNANI e JOHN, 2001). Isto implica na detenção do processo de projeto para se dedicar a atividades descritivas desnecessárias nas etapas iniciais.

¹⁹ Protocolos de uso, como o de definição de janelas exemplificado na página anterior.

3. **Diminui a qualidade do projeto:** em muitos casos, o projeto resultante de um processo iniciado em CAD obtém uma avaliação baixa em comparação com resultados de projetos iniciados com lápis e papel, como tem sido registrado por professores de Projetos Arquitetônicos (ASANOWICZ, 2003) (IBRAHIM e RAHIMIAN, 2010) (DENZER e GARDZELEWSKI, 2011). Os avaliadores identificam uma restrição no universo de soluções a aquelas que um determinado software CAD suporta, ou aquelas que o projetista soube produzir com esta ferramenta, privando-o de produzir melhores soluções.

Das três razões acima descritas, pode-se concluir que a tendência de usar ferramentas CAD nas etapas iniciais de projeto faz deste um processo lento, ineficiente, pouco comunicativo da abordagem de solução, e com um resultado que tende a ser pior do que os projetos iniciados com sketches à mão livre (ASANOWICZ, 2003) (IBRAHIM e RAHIMIAN, 2010) (DENZER e GARDZELEWSKI, 2011).

Para impedir as consequências acima descritas e, ao mesmo tempo, garantindo a continuidade das informações nas transições entre etapas de projeto, é necessário que, nas etapas iniciais, os desenhos sejam elaborados com ferramentas digitais que suportem a elaboração de sketches, assim como o lápis e o papel (JUCHMES, LECLERCQ e AZAR, 2005). Um sistema de computação como este contribuiria para atingir uma maior qualidade no projeto ao permitir uma elaboração mais rápida de ideias de partido mais fiéis ao imaginado pelo projetista, e oferecer um meio de comunicação mais informativo, através de uma interação mais apropriada do que a dos atuais softwares de CAD e modelagem em 3D (*ibid.*).

Diferentemente do AutoCAD e de softwares tradicionais de modelagem em 3D como Autodesk 3ds Max, Trimble SketchUp preconiza uma interação intuitiva ao usar metáforas de modelagem simplificadas para nomear funcionalidades (p. ex.: “*Follow-me*” em vez de “*Extrude Along Curve*”); diminuir a quantidade de botões; e simplificar os seus ícones (TRIMBLE NAVIGATION LTD., 2013). Mas em realidade, SketchUp oferece uma grande quantidade de funcionalidades cifradas em menus e barras de botões não visíveis por default, complicando a sua utilização. No uso representativo deste software em projetos de arquitetura, o projetista tende a definir a aparência dos materiais de acabamento junto com a definição geométrica do modelo. Este software provê mais suporte para as etapas avançadas de projeto do que para as etapas iniciais (PRANOVICH, 2004).

1.2.4 Interação das simulações de desempenho

Entre o suporte oferecido através de SketchUp, encontra-se a vinculação a sistemas de análises ambientais, como o *plug-in* de OpenStudio para SketchUp. Com este *plug-in*, é possível usar o modelo 3D como dado de entrada para as análises do OpenStudio, apoiado no motor de cálculos ambientais EnergyPlus²⁰. Além do uso para input, OpenStudio oferece uma funcionalidade de geração automática de alternativas de fachadas, vinculando as características do modelo aos parâmetros da análise. Esta funcionalidade não inclui uma visualização dos resultados da análise através de simulações de desempenho. Os resultados são apresentados em formato de relatórios de texto, tabelas e gráficos, que não são visualmente explícitos em relação à situação do desempenho do projeto em elaboração (CRAWLEY, HAND *et al.*, 2008). O projetista depende da sua capacidade de interpretação de resultados de desempenho para relacioná-los com o modelo e identificar nele os problemas de desempenho levantados pela análise.

Além de OpenStudio, existem softwares de análise ambiental direcionados para arquitetos, como Autodesk Ecotect e Autodesk Vasari. Ecotect e Vasari oferecem funcionalidades, não só de análise, mas também de simulação de desempenho. Entre os fatores ambientais sobre os quais Ecotect oferece funcionalidades de simulação, encontram-se: desempenho térmico, radiação solar, impacto visual, sombras e reflexões, e iluminação natural. Para todas estas simulações, Ecotect gera uma malha poligonal discretizada, como a descrita inicialmente²¹. A interação que o projetista tem com esta simulação não é imediata. O modelo deve ser preparado, seja modelando diretamente no Ecotect ou importando um modelo elaborado em outros softwares, como SketchUp ou Rhinoceros (VANNINI, 2011). Esta preparação também sofre dos problemas de distanciamento do foco da atividade, como foi previamente levantado, com o qual a *affordance* que estas simulações apresentam com relação ao potencial de melhoria do desempenho do modelo não são suficientemente exploradas (GRABNER e FRICK, 2013)²².

Vannini (2011) estudou uma aplicação tecnológica de softwares que vincula automaticamente a simulação de desempenho de radiação solar do Ecotect com modelagem

²⁰ Tanto EnergyPlus como OpenStudio são softwares desenvolvidos pelo Departamento de Energia dos EUA. **EnergyPlus Energy Simulation Software**. Disponível em: <<http://apps1.eere.energy.gov/buildings/energyplus/>>. Acesso em 17 jun 2014. NREL: **OpenStudio**. Disponível em: <<https://openstudio.nrel.gov/>> Acesso em 17 jun 2014.

²¹ Veja-se descrição da discretização de modelos 3D para simulação de desempenho na página 7 da Introdução.

²² Grabner e Frick (2013) falam em perda do potencial aproveitamento das funcionalidades da simulação. É nossa a vinculação desses conceitos para formular a afirmação sobre o mal aproveitamento da *affordance* do modelo através das simulações.

paramétrica e otimização por algoritmos evolutivos em Grasshopper, um aplicativo de modelagem algorítmica para McNeel Rhinoceros. Esta vinculação automática tem o potencial para encurtar a interação cíclica entre o projetista e o modelo simulado (GRABNER e FRICK, 2013), mas a otimização evolutiva modifica a natureza da interação ao substituir parte das atividades de edição direta do modelo pela automatização da avaliação preliminar de milhares de possíveis soluções. Por meio do domínio de habilidades de modelagem algorítmica e controle de parâmetros de computação evolutiva, o projetista se apoia nas capacidades de processamento do computador para aumentar as suas habilidades projetuais (RUTTEN, 2013). Entre os problemas levantados por Vannini (2011), destaca-se a grande quantidade de recursos de computação que consome este procedimento (uso da memória do computador e tempo de execução), levando a atrasos na elaboração do projeto, fazendo-o inviável para geometrias não primitivas. Isto pela ineficiência da execução dos cálculos e simulações do Ecotect, solicitados uma e outra vez para cada iteração do algoritmo evolutivo (*ibid.*). Vannini, Bueno e Turkienicz (2012) apresentam um avanço na solução deste problema ao dispensar o uso do Ecotect neste procedimento e incluir as funcionalidades de análise e simulação dentro da programação da modelagem em Grasshopper. O ganho na eficiência do uso dos recursos de processamento permite apreciar a aproximação do projetista na interação cíclica com o modelo simulado. Porém, ainda não incide de uma maneira suficientemente direta na tomada de decisões de projeto, que, neste caso, é feita indiretamente pelo projetista na configuração dos parâmetros de automatização. A interação entre projetista e projeto é mais próxima da interação CAD do que de uma interação própria das atividades de projeção nas etapas iniciais, tipicamente instrumentada por lápis e papel.

Autodesk Vasari também apresenta uma funcionalidade que supera as limitações do Ecotect ao simplificar as definições geométricas que utiliza internamente para a elaboração dos cálculos para as análises de desempenho, e ao oferecer uma interface gráfica mais simples (NOME e QUEIROZ, 2013). Estudos assinalam este software como o mais apropriado dentre os softwares de análise de desempenho ambiental para as etapas iniciais de projeto por ser o mais simples e abordar as análises da maneira mais geral possível (*ibid.*). Mas a necessidade de especificar atributos de modelagem para definir o seu formato BIM se interpõe entre as *affordances* das simulações (a percepção de possibilidade de aproveitamento das simulações) para direcionar as tomadas de decisão de projeto. Ainda, esta especificidade afasta as *affordances* das simulações das atividades aqui descritas como as mais importantes nas etapas iniciais de projeto: a elaboração de ideias de partido através do desenho de sketches.

Para conseguir uma aplicação tecnológica que integre funcionalidades de desenho de sketches com funcionalidades de simulação de desempenho, uma evolução do software para implementar estas funcionalidades integradas não seria suficiente se o dispositivo de entrada para a aquisição do sketch continuar a ser o mouse. As inconsistências na *affordance* do mouse (BUXTON, 1986) descritas na Introdução²³ justificam a consideração de outros modos de interação.

No campo da IHC, **modo** é o termo que se usa para definir um canal de comunicação humano-computador (DIX, FINLAY *et al.*, 2003). Além dos modos de interação WIMP, que privilegiam a postura WYSIWYG, dispositivos como as webcams adicionaram a entrada de imagens, vídeo e som que, através de processos de reconhecimento de padrões, podem definir gestos como comandos de entrada, que aumentam a capacidade comunicativa do computador (desde que o usuário considere sua parte desta comunicação na postura WYSIWYM). Isto define o início do paradigma da multimodalidade, especialmente o uso simultâneo de múltiplos canais para entrada e saída (*op. cit.*).

Do paradigma da multimodalidade, é especialmente relevante nesta pesquisa a consideração de modos de entrada direta através de dispositivos como canetas e telas sensíveis ao toque.

1.2.5 Evolução da interação por caneta e *touch*

Sobre sistemas de interação através do desenho direto na tela do computador, existem registros desde 1956 (JOHNSON, 1956 *apud* BUXTON, 2013). O princípio inicial da tecnologia de telas sensíveis ao toque (ou telas *touch*), ainda vigente em alguns sistemas de hardware (BUXTON, 2013), se baseia na utilização de sensores capacitivos integrados na tela para captar a eletrostática do dedo humano no contato com a tela. Estes sensores estão arrançados como uma matriz, na qual cada um deles possui coordenadas nas duas dimensões, que são informadas ao software gráfico no momento em que o usuário ativa o sensor correspondente ao tocar a tela no espaço acima deste sensor (*ibid.*).

No que diz respeito ao desenho por caneta, Sutherland (1963) conseguiu a primeira abordagem relevante ao problema do desenho computadorizado com interação direta na tela, evitando as limitações tecnológicas dos sistemas *touch* da época. Isto foi conseguido no desenvolvimento do Sketchpad, sistema de software e hardware para desenhar, que implementava um sistema de interação com entrada por caneta de luz. A entrada era

²³ Veja-se a descrição da *affordance* do mouse, na Introdução, página 8.

complementada com um conjunto de botões e interruptores que instruíam o computador como interpretar o movimento da caneta de maneiras diferentes. Os interruptores ativavam e desativavam funções de criação de linhas e polígonos, edição desta geometria, transformações geométricas e definição de restrições geométricas, como por exemplo, relações de paralelismo entre objetos desenhados (Computer Sketchpad (Science Reporter), 1963). Mesmo considerando a interface física de botões e interruptores, é conhecido como o primeiro projeto de IHC em apresentar uma interface gráfica de usuário (REAS, MCWILLIAMS e BARENDSE, 2010).

Pesquisas posteriores, como a de Weinzapfel, Johnson e Perkins (1971), utilizaram o Sketchpad como base de interação e motor de desenho sobre o qual construíram um software de assistência às etapas iniciais de projeto, denominado IMAGE. Este software estendia as funcionalidades de controle de restrições geométricas para restrições entre as relações de conjunto do *layout* arquitetônico e urbano (*ibid.*).

Outra incipiente aplicação computacional para assistência a atividades criativas, o terminal do sistema de computador PLATO IV, apresentou a invenção da tela plana de plasma para oferecer um sistema de interação *touch* acessível²⁴ (SHERWOOD e STIFLE, 1975). Pelas restrições impostas pelo contexto de aplicação (um sistema de apoio ao ensino auxiliado por computador com distribuição em escolas públicas na década de 1970), o terminal contava com uma tela de tamanho reduzido e o espaço de interação de baixa resolução (matriz de 16×16 pontos de toque) (BUXTON, 2013). Isso fazia da tela um dispositivo apropriado para funções de entrada de pontos (como em respostas de seleção simples relevantes em pedagogia), mas não para funções de desenho ou edição de gráficos.

Herot e Weinzapfel (1978) definiram soluções com as quais é possível utilizar o toque com os dedos como entrada de desenho e controle de manipulação de elementos gráficos. O sistema de entrada por toque de um ponto implementava na tela ao contato com o dedo do usuário, um digitalizador sensível a pressão. O sistema aproveitava a fricção entre o dedo e o vidro para detectar forças de pressão não só perpendiculares à tela, mas também forças laterais e de torque. Esta ampla capacidade de captação era aproveitada para definir cinco funcionalidades: cursor de força (definição visual de um vetor segundo a orientação do dedo em contato com a tela), histórico vetorial (registrar em linha poligonal a trajetória de um vetor que era guiado com o dedo estacionário), empurrar/puxar, dispersão e rotação. Os resultados

²⁴ Neste caso, a tecnologia *touch* utilizada era baseada num feixe de luz infravermelha de barrido na tela (SHERWOOD e STIFLE, 1975). Assim como os sensores capacitivos, esta tecnologia também é utilizada até hoje (BUXTON, 2013).

demonstraram que é possível trabalhar com uma alta precisão processando pouca informação. Mas a intenção de prover um sistema de interação intuitiva fica limitado ao toque de um só dedo (*ibid.*).

Sistemas de detecção de mais de um toque simultaneamente, ou *multi-touch*, começaram com grandes dispositivos que combinavam retroprojeção e captura de vídeo dentro de uma mesa cujo tampo era a tela, ou *tabletop* (MEHTA, 1982). Através destes sistemas de entrada e saída de vídeo digital²⁵, era possível oferecer funcionalidades para atividades de escritórios e ateliês de trabalhos visuais – incluindo o desenho – contidas nesta mesa aumentada, na qual o trabalho é criado digitalmente desde o seu início (NAKATANI e ROHRLICH, 1983). Estes avanços complementam fisicamente a metáfora visual de trabalho em escritório inerente aos elementos WIMP. Adicionalmente, pesquisas como a de Krueger (1983) (KRUEGER, GIONFRIDDO e HINRICHSEN, 1985) começaram a explorar os gestos, feitos pelas mãos dos usuários nesta interação, como disparadores diretos das ações que levam à elaboração do trabalho no computador. Desta maneira, definiram um repertório de gestos que sentaram as bases dos atuais vocabulários de gestos *multi-touch* (BUXTON, 2013).

Durante as décadas de 1980 e 1990, a comunidade científica presenciou uma diversidade de pesquisas e desenvolvimento de protótipos baseados nos avanços acima discutidos, dos quais, os mais significativos são descritos em Buxton (2013). Estes trabalhos, mesmo envolvendo colaborações entre a academia e a indústria, não geraram ferramentas para usuários finais que incidissem nas atividades profissionais (ELSEN, DARSES e LECLERCQ, 2011). Uma exceção é o trabalho em conjunto entre a Universidade de Toronto, Alias|Wavefront e Wacom (LEGANCHUK, ZHAI e BUXTON, 1998), utilizando o sistema de entrada com base em *tablets* opacas Wacom para estudar os benefícios do trabalho simultâneo com caneta e *puck*²⁶, e definir critérios de uso de entradas bi-manuais (*ibid.*).

Na década de 2000, se registrou uma prolífica produção acadêmica sobre softwares que, contando com dispositivos de entrada direta, especialmente canetas *stylus*, incursionaram no reconhecimento e interpretação de sketches para desenho e modelagem por reconstrução geométrica. Aplicados a projetos, estes conceitos se referem a: **reconhecimento** de elementos geométricos desenhados; **interpretação** deles como elementos de projeto; e **reconstrução** de

²⁵ Considerando a utilização de vídeo digital nestes sistemas de entrada, o registro da interação era feito diretamente no software, sem ter que processar sinais do mouse o teclado, denominados *soft machines* (NAKATANI e ROHRLICH, 1983).

²⁶ O *puck* é um dispositivo similar ao mouse em aparência, mas em vez mapear as coordenadas relativas do movimento, trabalha com coordenadas absolutas da sua posição e orientação dentro do espaço da mesa digitalizadora (BUXTON, 2013). Não se registram aplicações recentes que utilizem o *puck*.

um modelo composto por estes elementos para a sua operação nas seguintes etapas de projeto (OLSEN, SAMAVATI *et al.*, 2009). Estudos comparativos desta produção têm sido feitos por Company, Piquer e Contero (2004), e por Olsen, Samavati *et al.* (2009). Entre os trabalhos estudados nestes levantamentos, que abordam o aproveitamento da entrada por caneta nas etapas iniciais de projeto ajudando na geração de modelos compatíveis com sistemas CAD, podem ser citados: Fonseca e Jorge (2001); Contero, Naya *et al.* (2003); Company, Contero *et al.* (2003); Leclercq (2004); Juchmes, Leclercq e Azar (2004); Masry e Lipson (2005); Contero, Naya *et al.* (2005).

No levantamento de Company, Piquer e Contero (2004) identifica-se na automatização do reconhecimento e interpretação de sketches prescritivos uma tendência ao desenvolvimento de uma percepção visual artificial, e conclui-se que a reconstrução 3D, nas áreas de projeto, tem o potencial para definir padrões de trabalho que simplificariam o exercício da profissão ao resgatar e unificar os elementos gráficos do desenho técnico tradicional, não só para a representação, mas também para o processo de desenho.

O estudo de Olsen, Samavati *et al.* (2009) foi mais atualizado. Evidencia que existem muitas limitações (interface, qualidade do modelo, precisão) ainda não ultrapassadas pelos trabalhos acima citados, especialmente as provenientes da falta de consideração com sketches que abordam problemas mal definidos. Os autores propõem que futuros avanços no desenvolvimento de sistemas de modelagem baseada em sketch combinem métodos automáticos de interpretação e reconstrução, com métodos interativos, nos quais o projetista possa direcionar o computador na construção do modelo. Por exemplo, que o projetista desenhe um sketch de um sólido simples em perspectiva, que o software reconstrua automaticamente, e logo, o modelo 3D possa continuar sendo editado por meio de transformações definidas por gestos. Os autores recalcam a importância do software ser estável quanto ao desempenho do computador e ter um comportamento tão previsível quanto o do lápis e papel (*ibid.*).

O comportamento previsível é um dos fatores importantes para garantir uma interação mais natural do que a tradicional interação WIMP. Esta é uma questão abordada em estudos de usabilidade. Um dos estudos de usabilidade mais relevantes neste campo é o desenvolvido por Company, Contero *et al.* (2006), com relação ao desenho de sketches em computadores *tablet*. Neste estudo, a interação com a caneta e a tecnologia de *tablets* então disponível foi valorada pelos usuários como desconfortante para a mão em comparação com o lápis e o papel. Especialmente, evidenciou-se no hardware disponível naquele momento (2006) um distanciamento visual entre a ponta da caneta e o ponteiro na tela que dificultava a resposta imediata entre o movimento da mão e o traço adquirido e visualizado. Isto contribuiu a concluir

que, na elaboração de sketches, a interação por caneta possuía maior usabilidade que a interação por mouse, mas que não era tão “usável” quanto o lápis e o papel (*ibid.*).

Considerando a usabilidade e efetividade das mídias tradicionais, Forlines, Wigdor *et al.* (2007) (BRANDL, FORLINES *et al.*, 2008) conduziram pesquisas nas quais experimentaram *tabletops* em combinações de modos de entrada com uma e duas mãos. Concluíram que o mouse é mais preciso para tarefas apontar e selecionar quando se usa uma só mão, porém apresenta maiores problemas de fadiga muscular na mão (FORLINES, WIGDOR *et al.*, 2007). Mas, com ambas as mãos, a melhor combinação foi a de caneta na mão dominante e toque com os dedos na outra mão (BRANDL, FORLINES *et al.*, 2008). Este resultado confirma a teoria de Guiard (1987) sobre a divisão assimétrica das tarefas elaboradas com cada mão no trabalho bimanual: a mão dominante pode trabalhar com muita precisão, e o trabalho pode ser mais eficiente e eficaz quando apoiado com ações complementares da mão não dominante de maneira simultânea.

1.2.6 Interação multimodal relevante para o processo de projeto

Hinckley, Yatani *et al.* (2010a) (2010b) desenvolveram uma pesquisa em torno a um protótipo de software denominado Manual Deskterity (HINCKLEY, YATANI *et al.*, 2010a) para funcionar como um caderno de notas digital para o *tabletop* Microsoft Surface (hoje denominado Microsoft PixelSense) (MUSIL, 2012).

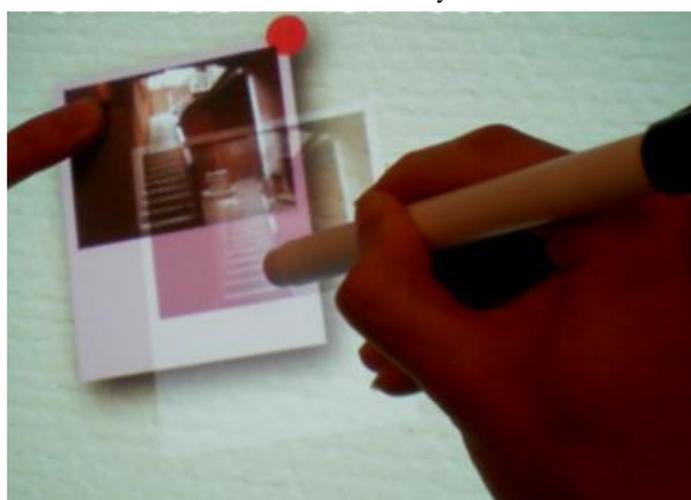
O avanço tecnológico de hardware aportado pela Microsoft Research no desenvolvimento do PixelSense foi incorporado na pesquisa de software de Hinckley, Yatani *et al.* (2010a) (2010b). Nessa pesquisa, estudaram um grupo de pessoas nas atividades de desenho, recorte e montagem para a elaboração de um design conceitual (um *storyboard*) num bloco de notas de papel. Foram observados os diferentes comportamentos e foram identificados os gestos associados que usaram para definir as funções que assistam um usuário de *tabletop*, considerando os critérios de divisão de tarefas de Guiard (1987) e de interação bimanual de Brandl, Forlines *et al.* (2008), executando as mesmas atividades de design conceitual com ferramentas digitais (HINCKLEY, YATANI *et al.*, 2010a).

Eles confirmaram que existe uma separação de funções entre as ações que são feitas com lápis ou caneta, e as que são feitas diretamente com os dedos. A caneta, quando utilizada na mão dominante, tem a vocação da criação gráfica com precisão, seja para escrever ou desenhar. A utilização direta dos dedos possui um controle psicomotor menos fino do que na utilização de caneta, porém suficientemente rápido e preciso para manipular objetos e folhas de papel. Basicamente, com a caneta se escreve e desenha, e com os dedos se manipula a

informação. A isto adicionaram uma terceira categoria definida pela interação conjunta das duas mãos: uma delas utilizando um instrumento e a outra usando os dedos; atribuída às tarefas de edição da informação, como recortar (HINCKLEY, YATANI *et al.*, 2010b).

No ambiente digital, esta diferença de funções se traduz em diferentes modos de iteração: “unimodal”: seja através de caneta *stylus* ou através dos dedos no contato com a tela *touch* ou *multi-touch*; e “multimodal”: através da utilização simultânea de caneta e *touch*, como o gesto mostrado na Figura 8 para duplicar um elemento gráfico (*op. cit.*).

Figura 8: Duplicação de um gráfico por meio de um gesto que combina entrada por caneta e *touch*, usando Manual Deskterity.



Fonte: Hinckley, Yatani *et al.* (2010a).

Os resultados de Hinckley, Yatani *et al.* (2010a) (2010b) demonstram que os usuários estudados, que representam uma população heterogênea que inclui os projetistas, conseguiram elaborar as tarefas de ideação no ambiente digital (no caso, um *storyboard*) com uma naturalidade comparável à atingida no trabalho com lápis, papel, tesouras e cola. Os usuários reportaram que não foi preciso nenhum esforço para aprender a empregar os gestos de edição. Isto permite um trabalho criativo muito mais fluido do que o possível com as tecnologias informáticas que trabalham com uma entrada por teclado e mouse.

Os resultados desta experiência reportam um incremento na usabilidade, ultrapassando barreiras identificadas por Company, Contero *et al.* (2006). A tecnologia de *tabletop* PixelSense diminui a distância entre a ponta da caneta em contato com superfície exterior da tela e o ponteiro no pixel ativo da tela, garantindo a correta percepção da aquisição do sketch e uma velocidade de resposta apropriada.

1.2.7 Evolução da modelagem baseada em sketch

Contero, Naya *et al.* (2003) e Naya, Conseca *et al.* (2003), desenvolveram um software experimental para modelagem em 3D denominado CIGRO, que permite utilizar uma caneta *stylus* para interagir diretamente na tela do computador, no desenho de sketches que são interpretados em perspectiva isométrica e reconstruídos em 3D. Na continuidade dessa pesquisa, Company, Contero *et al.* (2003), desenvolveram REFER, software que introduz métodos alternativos de interpretação e reconstrução do sketch. Posteriormente, o software GEGROSS (NAYA, CONTERO *et al.*, 2004), apresentado na Figura 9, estende a funcionalidade do CIGRO, permitindo interação por meio de gestos, demonstrando ganhos em usabilidade e eficiência no que se refere à interação por caneta *stylus* (NAYA, CONTERO *et al.*, 2008). Estes softwares não suportavam interação *multi-touch*.

Figura 9: Usuário desenhando um sketch com uma caneta *stylus* num computador *tablet* como entrada de modelagem no software GREGROSS.



Fonte: Naya, Contero *et al.* (2008).

Mais recentemente, Kang, Kim *et al.* (2013) desenvolveram um protótipo de aplicativo para computadores *tablet* de modelagem baseada em características (modelagem paramétrica em função da relação de características de peças mecânicas, ou MCAD) que aproveita a interação *multi-touch*, implementando funções de modelagem baseada em sketch controlada por gestos, similar a Naya, Contero *et al.* (2008), mas considerando a baixa resolução da interação direta dos dedos, sem suporte para caneta. Além da implementação de tecnologias de software mais intuitivas para o desenho de sketches, oferece interoperabilidade com os softwares de MCAD comerciais (KANG, KIM *et al.*, 2013). Como colocado por Olsen, Samavati *et al.* (2009), estes esforços por conseguir uma interação intuitiva são cruciais para a

adoção destas tecnologias. Na evolução da pesquisa sobre interação entre o projetista e a tecnologia de projeto, inicialmente, o lápis se demonstrava o mais intuitivo. Em segundo lugar, estavam os softwares CAD na utilização de canetas e *tablets*, e, por último, estes softwares com interação de teclado e mouse. Esta diferença se demonstrou mais acentuada em projetistas experientes do que em jovens projetistas (COMPANY, CONTERO *et al.*, 2006). Inclusive, a relação é mais distante em termos de eficiência e eficácia: os projetistas experientes são mais rápidos elaborando melhores sketches (segundo avaliação de desenho técnico) a lápis e papel do que os jovens projetistas (*ibid.*). Isto ressalta a importância da experiência prática do projetista na sua percepção construtiva sobre os métodos permitidos pelas ferramentas de projeto e sobre a usabilidade da ferramenta em si, fator determinante para a adoção de uma determinada tecnologia em projetos.

1.3 ADOÇÃO DE TECNOLOGIAS DE CANETA E *MULTI-TOUCH*

A tecnologia *multi-touch* já é acessível ao usuário final de produtos eletrônicos de consumo e de equipamentos de escritório (INGRAHAM, 2013). A atual tecnologia tem se popularizado desde 2007, quando Apple começou a comercialização do iPhone (BUXTON, 2013), e mais ainda desde 2010, quando a empresa lançou o computador *tablet* iPad (INGRAHAM, 2013) e outras empresas lançaram produtos similares, como por exemplo, o Galaxy Tab, da Samsung (HAMBLEN, 2010). A atual tecnologia *multi-touch* é resultado de uma evolução que tem levado décadas para amadurecer. Isto vem acontecendo na maioria das inovações tecnológicas²⁷ (BUXTON, 2008), o que provê, neste estágio de amadurecimento, certa coerência nas melhoras incrementais que respondem às necessidades dos usuários e garante a suficiente estabilidade da categoria de produtos, como para ser considerada como ferramenta de trabalho por toda uma classe profissional (*ibid.*).

1.3.1 Adoção de canetas e *multi-touch* em outras disciplinas criativas

Fora da área da arquitetura, as tecnologias de caneta e tela *multi-touch* apresentam alto grau de adoção em outras disciplinas criativas, como a utilização de iPads nas artes visuais e a ilustração (SAORÍN, DE LA TORRE *et al.*, 2011). Além das pesquisas acadêmicas (*ibid.*), esta adoção é evidente na grande presença de grupos de artistas visuais e designers gráficos, usuários destas tecnologias. Os usuários se organizam em redes sociais e profissionais, como as criadas em torno do uso do software FiftyThree Paper ou do Autodesk SketchBook (DICHERRI, 2013).

²⁷ “A maior parte da inovação é de baixa amplitude e (...) ocorre ao longo de um longo período de tempo” (BUXTON, 2008).

Nestas redes, os usuários contam com galerias para publicar as suas imagens, divulgando avanços profissionais e demonstrando confiança na adoção destas tecnologias; contam com fóruns, nos quais podem responder a dúvidas de outros usuários, contribuindo no aumento na adoção destas tecnologias. O alto grau de adoção atual também é consequência de uma consistência no suporte de funcionalidades que os desenvolvedores de softwares direcionados às artes visuais e design gráfico têm oferecido há anos. Além do suporte específico a entrada por caneta *stylus* em Adobe Photoshop para a aplicação de pinceladas (LEGASPI, 2010), Adobe Illustrator suporta a entrada por caneta desde 2005. Observa-se, também, um direcionamento ao desenvolvimento de aplicativos para editoração de imagens, com interação *multi-touch* para computadores *tablets*, como na evolução do Autodesk SketchBook (ANSARI, 2009) e no Adobe Photoshop *Touch* (DOVE, 2012). Os artistas e designers gráficos que instrumentam seu trabalho com softwares como os de Adobe ou Autodesk mencionados aumentam sua produtividade com o passar dos anos ao acompanhar a atualização que estes softwares vêm tendo, especialmente no suporte de entrada por caneta e gestos *multi-touch*. Estas tecnologias de entrada fazem o trabalho de artes visuais e ilustração mais intuitivo que com a entrada por mouse e teclado, por serem a caneta *stylus* e gestos *multi-touch* mais próximos ao trabalho manual (com lápis, pinceis, etc.) que fundamenta as técnicas de expressão nestas disciplinas (SAORÍN, DE LA TORRE *et al.*, 2011).

1.3.2 Falta de adoção em projetos de arquitetura e urbanismo

Em contraposição, em projetos de arquitetura e urbanismo, estas tecnologias de caneta e *multi-touch* não apresentam um alto grau de adoção (ELSEN, DARSES e LECLERCQ, 2011). Como mencionado anteriormente, o uso predominante que os arquitetos fazem dos computadores para projetar, se limita a computadores pessoais (desktop ou laptop) instrumentados com entrada por teclado e mouse. Em relação à exploração de dispositivos de entrada direta, controlando os softwares anteriormente mencionados, Vizioli, Lancha e Castral (2013) investigaram o desenho digital à mão livre relacionado à atividade projetual na arquitetura e design de produto. Utilizaram canetas *stylus* sobre *tablets* opacas (ou mesas digitalizadoras) em computadores pessoais (notebook ou laptop) para o desenho de sketches em Adobe Photoshop na elaboração de designs conceituais de luminárias. Observaram que diferentes métodos de redesenho e re-representação são possíveis, permitindo a reinterpretação dos sketches nas tomadas de decisão de projeto. Concluíram que esta aplicação tecnológica expande as possibilidades de exploração conceitual, “preservando as características fundamentais do desenho à mão livre, como a imprecisão e proximidade ao pensamento do

projetista”, ao mesmo tempo em que aproveita recursos cognitivos pertencentes às mídias digitais. Associado à personalidade do projetista estampada na ambiguidade do traço desenhado, o olhar continua a ser basilar no processo de projeto (VIZIOLI, LANCHI e CASTRAL, 2013).

Figueredo e Vizioli (2013) ampliaram este estudo, comparando o mesmo tema projetual (design conceitual de luminárias) elaborado com diferentes aplicações tecnológicas, definidas pela configuração de hardware: *tablet* opaca Wacom Bamboo; computador *tablet* Apple iPad; tela *multi-touch* Wacom Cintiq; caneta eletrônica Wacom Inkling; e lápis e papel, como instrumentos do método de controle. Observaram que a caneta eletrônica é usada de maneira diferente às mídias tradicionais, uma vez que o usuário está ciente que os seus traços estão sendo gravados e serão reproduzidos como uma animação. Isto direcionou o projetista a um sketch menos ambíguo. A *tablet* opaca possui a mais inclinada curva de aprendizagem no início, mas ofereceu maior precisão do que o computador *tablet*. Este último ofereceu um melhor feedback visual dos movimentos da caneta, mas se registrou limitado no tamanho da tela em relação ao tamanho apropriado para o tipo de desenho. Por não depender de um computador pessoal (desktop ou laptop), ofereceu a melhor portabilidade. A tela *multi-touch* apresentou a melhor combinação de benefícios: uma naturalidade de uso similar à das mídias tradicionais e à da caneta eletrônica, uma precisão similar à da *tablet* opaca, e o mesmo feedback visual do computador *tablet*, oferecendo uma tela suficientemente grande para o tipo de desenho necessário, e melhor aproveitando a amplitude de movimento do braço do usuário (FIGUEREDO e VIZIOLI, 2013).

Devido aos softwares utilizados nestas experiências (VIZIOLI, LANCHI e CASTRAL, 2013) (FIGUEREDO e VIZIOLI, 2013), os sketches resultantes são imagens em mapa de bits. Na necessidade de continuar o desenvolvimento desses projetos nas etapas avançadas de projeto, que demandam de desenhos vetoriais, estas imagens devem passar por um processo de interpretação das ambiguidades mencionadas nos sketches antes da vetorização, tema não abordado pelos autores. Na ausência deste suporte, a assistência que estas aplicações oferecem ao processo de projeto de arquitetura (inclusive ao de design de produto) é insuficiente. Softwares de editoração de imagens como Adobe Photoshop, Corel Photo-Paint ou FiftyThree Paper (utilizado por Figueredo e Vilioli (2013) no experimento com iPad) não produzem desenhos vetoriais, porém, não são diretamente utilizáveis no desenho CAD. A transição implica em procedimentos como os descritos na transferência a partir de sketches em papel,

digitalizados²⁸. Softwares como Adobe Illustrator ou Corel Draw, mesmo gerando imagens vetoriais, não trabalham em 3D, nem oferecem interoperabilidade direta com os softwares CAD que possuem funcionalidades de modelagem em 3D.

Softwares que abordam a entrada de caneta e tela *multi-touch* com suporte à modelagem em 3D para projetos de arquitetura também têm sido alvo de pesquisas acadêmicas (JUCHMES, LECLERCQ e AZAR, 2004) (YU e ZHANG, 2007) (ALAÇAM e ÇAĞDAŞ, 2008) (RIGHI e CELANI, 2008) e da indústria de software. Autodesk desenvolveu, em 2002, o software Architectural Studio com o intuito de atingir a assistência nas etapas iniciais de projetos arquitetônicos e a transição às etapas avançadas. Numa interface familiar para o arquiteto, o software suportava desenho de sketches sobreposto em camadas acima de imagens, desenhos CAD e modelos em 3D, especialmente para o trabalho básico do contexto espacial do projeto, e a elaboração de pranchas de projeto (CHRABIN, NEUCKERMANS e SZEWCZYK, 2003) (PRANOVICH, 2004). Oferecia interoperabilidade com softwares CAD e 3D, mas não contava com interpretação dos sketches. O software foi descontinuado, e não se acham registros sobre o seu uso após 2004, ou sobre as razões da sua descontinuação.

No lançamento do sistema operacional Windows 7, Microsoft formou uma parceria com SpaceClaim para integrar neste software de modelagem de sólidos o suporte completo de interação por caneta e *multi-touch* (SPACECLAIM CORP., 2009). Este avanço foi considerado com bons resultados e levado em conta como antecedente em algumas pesquisas acadêmicas (KANG, KIM *et al.*, 2013), mas não se registra alguma prática representativa sobre o aproveitamento desta melhoria específica.

Para o nicho dos dispositivos móveis *multi-touch*, alguns dos principais desenvolvedores de soluções CAD, como Autodesk e Graphisoft, tem desenvolvido aplicativos específicos. Autodesk FormIt oferece funcionalidades de modelagem em 3D de sólidos primitivos num ambiente geo-referenciado e com simulação de sombreamento segundo a luz do sol para um lugar e momento determinado. Após o uso deste aplicativo, é possível afirmar que tem limitações importantes em relação às atividades levantadas nesta dissertação como relevantes nas etapas iniciais de projeto, já que não suporta sketches ou desenhos similares. Em vez disso, as formas são inseridas numa dimensão padrão e logo editadas. Na interação direta com os dedos, a geração e manipulação dos objetos se dificulta à medida que a agregação de formas torna o modelo complexo. Há menos de dois anos do seu lançamento, FormIt não só

²⁸ Veja-se a descrição da transferência de sketches em papel a desenhos CAD na Introdução, página 8.

conta com suporte para *tablets* iPad²⁹, existem também versões para dispositivos Android e uma versão Web que pode ser executada em computadores *desktop*, mesmo sem entrada *touch*³⁰.

Em geral, existe uma variedade de aplicativos relacionados com a indústria da arquitetura, engenharia e construção, mas somente Autodesk FormIt suporta modelagem preliminar em 3D, e o restante oferece funções de mera visualização. Desconhece-se a disponibilidade de algum aplicativo para dispositivos móveis *multi-touch* que assista na criação e edição de desenhos 3D a partir de sketches arquitetônicos (KANG, KIM *et al.*, 2013). No suposto da adoção massiva destes aplicativos de visualização na arquitetura, ainda não estariam sendo aproveitadas as tecnologias de caneta *stylus* e gestos *multi-touch* no suporte às atividades mais importantes da elaboração de projetos de arquitetura.

Resumidamente, expõem-se os argumentos da falta de adoção destas tecnologias de caneta e *multi-touch* em duas razões fundamentais:

1. Entre os softwares relevantes disponíveis a usuários finais, incluindo os que suportam interação com canetas e *multi-touch*, não se encontra algum que dê suporte à elaboração de sketches de natureza difusa e que permitam ao projetista fazer uso da sua apurada inteligência visual para encontrar nos seus sketches formas emergentes e que, ao construir uma “re-representação” digital destes sketches, permita que estas ideias avancem às seguintes etapas de projeto de uma maneira fluida.
2. As vantagens do trabalho no ambiente digital³¹ não são suficientes para motivar a adoção destas tecnologias. Os trabalhos acadêmicos que têm apontado avanços nesta direção do software foram restringidos por problemas de usabilidade. Company, Contero *et al.* (2006) concluíram em 2006 que estas tecnologias só chegariam a ser amplamente adotadas quando forem percebidas pelos projetistas como possuindo uma facilidade de uso igual ou superior ao lápis e papel³², o que não aconteceu até 2010 (HINCKLEY, YATANI *et al.*, 2010a).

²⁹ AUTODESK INC. New Autodesk FormIt App Accelerates Conceptual Design and Form Creation for Building Design Professionals. **Autodesk Newsroom**. 27 nov. 2012. Disponível em: <<http://news.autodesk.com/press-release/new-autodesk-formit-app-accelerates-conceptual-design-and-form-creation-building-design>>. Acesso em: 5 mai. 2014.

³⁰ AUTODESK INC. Introducing FormIt Web Beta. **Autodesk FormIt**. 16 out. 2013. Disponível em: <<http://autodeskformit.com/2013/10/16/introducing-formit-web-beta/>>. Acesso em: 5 mai. 2014.

³¹ Vantagens do trabalho no ambiente digital: veja-se na página 10 da Introdução, no ponto Interação dos dispositivos de entrada para projetos de arquitetura.

³² Qualidades de usabilidade de caneta sobre *tablet* em 2006: veja-se a página 10 da Introdução.

Como pode ser visto no seguinte capítulo, existem, no atual estado da arte, algumas experiências desenvolvidas sobre hardwares que já superaram as limitações percebidas por Company, Contero *et al.* (2006), mas não combinam todos os avanços levantados, de maneira a aproveitar cabalmente estas tecnologias para a elaboração de projetos de arquitetura.

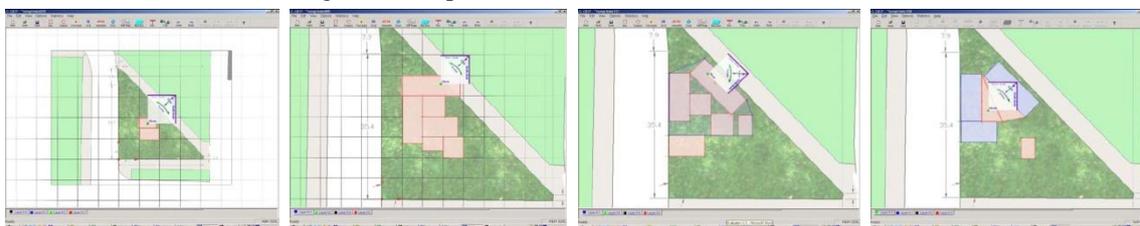
2 ESTADO DA ARTE

Muitos dos trabalhos acadêmicos citados no capítulo anterior, além de definir fundamentos das funcionalidades necessárias, têm aportado avanços no campo através da implementação destas funcionalidades em softwares experimentais. Destes softwares experimentais, se estudam treze (posteriormente descritos), dos quais oito são direcionados à arquitetura: Pranovich (2004), que desenvolve Structural Sketcher; Juchmes, Leclercq e Azar (2004), que apresentam EsQUIsE-SMA; Kallio (2005), que desenvolve o 3D6B Editor; Oh, Stuerzlinger e Danahy (2006), que apresentam SESAME; Yu e Zhang (2007), que apresentam o protótipo SmallBoom; Dorsey, Xu *et al.* (2007), que desenvolvem Mental Canvas; Elsen e Leclercq (2008), que desenvolvem SketSha; e Elsen, Demaret *et al.* (2012), que desenvolvem NEMo. Os cinco softwares restantes neste estudo não são direcionados à arquitetura, mas são incluídos devido ao potencial de aproveitamento das suas contribuições. São eles: REFER, por Company, Contero *et al.* (2003); o protótipo de Masry e Lipson (2005); GEGROSS, por Naya, Contero *et al.* (2008); ShapeShop Multitouch, por Lopes, Mendes *et al.* (2011); e o aplicativo móvel MPA, por Kang, Kim *et al.* (2013). Na sequência, descrevem-se as características destes softwares, agrupadas por categorias segundo a configuração geométrica explorada no uso do software, a interação permitida pelo sistema de entrada abordado, as funcionalidades de editoração geométrica e a sua compatibilidade com os sistemas operacionais relevantes para arquitetos.

2.1 CONFIGURAÇÃO GEOMÉTRICA TESTADA

Pranovich (2004) testa o Structural Sketcher com usuários projetando a planta de um pequeno consultório médico (Figura 10). Estes usuários criam espaços predominantemente ortogonais.

Figura 10: Capturas do teste do Structural Sketcher.



Fonte: Pranovich (2004).

O processo inicia estudando o programa de necessidades e gerando retângulos com os tamanhos dos espaços requeridos. Para controlar a escala dos espaços delimitados, o usuário se apoia em recursos gráficos como a grelha, pontos de conexão e transformações, e outros,

denominados unidades gráficas. Estas unidades gráficas são parametricamente associáveis às formas desenhadas, atuando como relações dos elementos de projeto (*op cit.*).

Em Juchmes, Leclerq e Azar (2004), utiliza-se uma casa unifamiliar projetada por um arquiteto, de geometrias predominantemente ortogonais combinadas com poucos planos inclinados (e um exemplo não explicado, que inclui uma curva radial) (Figura 11).

Figura 11: Janelas do EsQUIsE interpretando os sketches de um arquiteto.

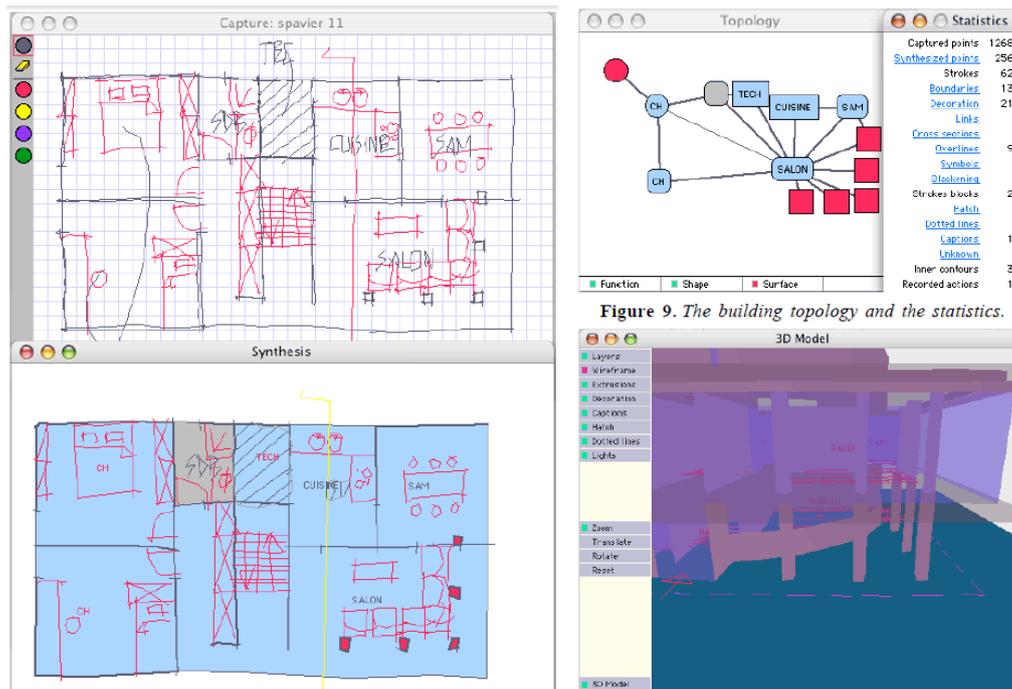


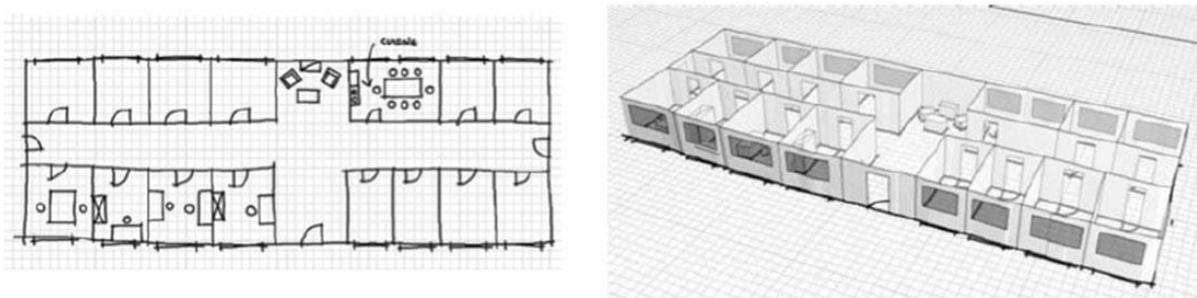
Figure 9. The building topology and the statistics.

Fonte: Juchmes, Leclerq e Azar (2004).

O processo de projeto testado em EsQUIsE, e ilustrado na Figura 11, começa com um sketch do layout da casa em planta. Na sequência é ilustrado o sketch reconhecido pelo software, a topologia dos espaços e relações interpretadas e o modelo 3D automaticamente gerado por extrusão das formas 2D reconhecidas. Os autores denominam esta aproximação de 2.5D, pela falta de definição geométrica tridimensional, que se limita à extrusão automática. Os autores justificam esta aproximação “reduzida”, já que atende uma grande parte das soluções arquitetônicas: pelo rol da gravidade, os elementos arquitetônicos estão forçados a ser predominantemente verticais ou horizontais, descritíveis com extrusões (*op cit.*).

Na menção ao software protótipo NEMo, Elsen, Demaret *et al.* (2008) apresentam um exemplo de uso, elaborando um pavimento ou planta tipo de uma edificação de escritórios, de geometrias totalmente ortogonais (Figura 12).

Figura 12: Interpretação de portas, janelas e mobiliário desenhado em NEMo.

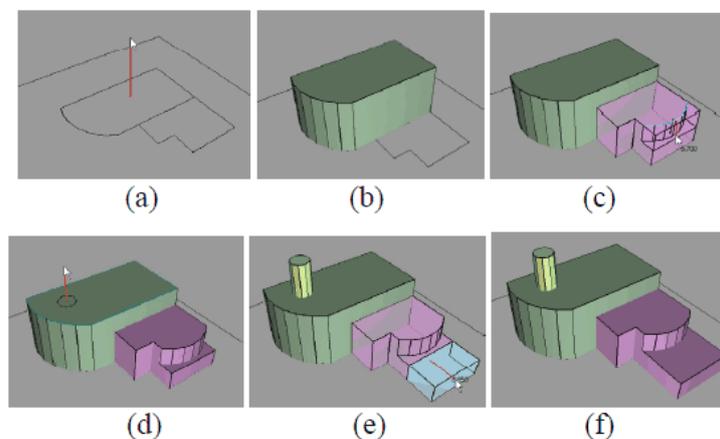


Fonte: Elsen, Demaret *et al.* (2008).

O processo demonstrado em NEMo inicia com um sketch em planta 2D. Na sequência, os autores apresentam o modelo 3D da planta tipo, na qual as paredes são automaticamente extrudadas. Adicionalmente, os símbolos da planta são interpretados como elementos arquitetônicos: portas, janelas e mobiliário, representados no modelo 3D.

Oh, Stuerzlinger e Danahy (2006), na explicação das funcionalidades de desenho e modelagem de SESAME, apresentam duas composições volumétricas de edificações. Os volumes destas composições são prismas, cujas bases são polígonos irregulares, com segmentos de linhas predominantemente retas (Figura 13).

Figura 13: Processo de construção geométrica em SESAME.

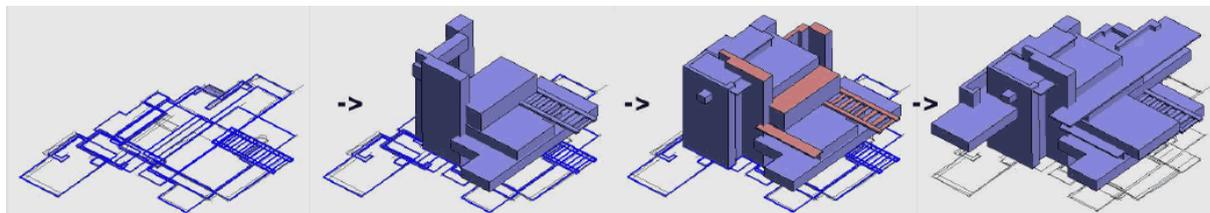


Fonte: Oh, Stuerzlinger e Danahy (2006)

Diretamente na vista em perspectiva, o usuário de SESAME elabora a composição arquitetônica traçando linhas 2D no plano horizontal, formando polígonos. Logo define um arrasto do ponteiro a partir de um ponto dentro de um polígono, para gerar o volume extrudado. Posteriormente, adiciona características geométricas mais específicas, como perfurações e saliências a partir das faces dos volumes gerados, utilizando a mesma sequência de desenho de polígonos e arrasto de extrusão ou protrusão, predominando a verticalidade das fachadas.

Yu e Zhang (2007) demonstram a utilização do protótipo SmallBoom, desenhando a Casa da Cascata, do Frank Lloyd Wright, uma obra prima da arquitetura cuja composição consta de volumes ortogonais (Figura 14).

Figura 14: Processo de modelagem baseada em sketch da Casa da Cascata em SmallBoom.

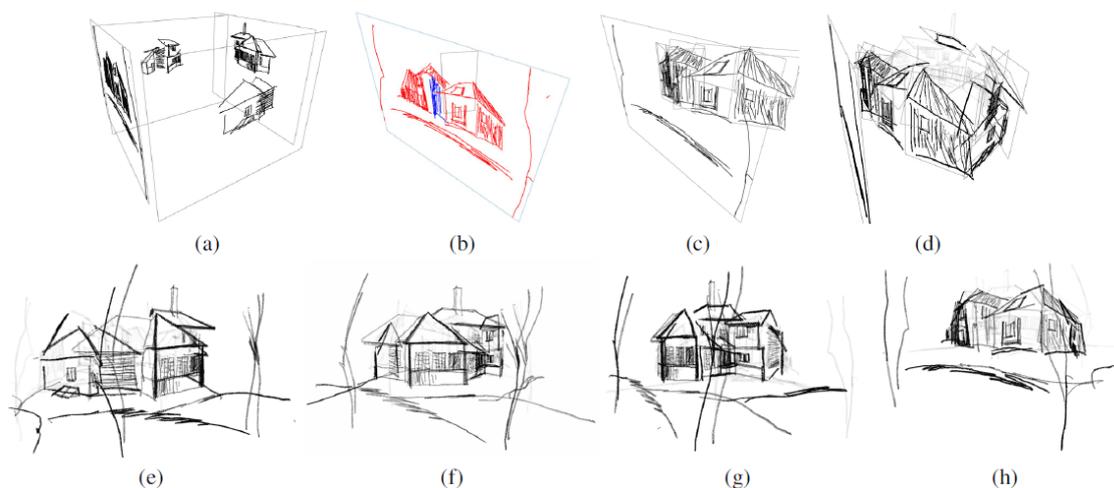


Fonte: Yu e Zhang (2007).

O usuário começa o processo de elaboração da casa fazendo traços de sketch no plano horizontal, formando retângulos para cada elemento arquitetônico. No momento em que o software reconhece um polígono fechado, oferece ao usuário a opção de definir uma espessura. Ao definir espessura, gera um volume extrudado que pode ser movido à determinada altura. Diferentemente de Oh Stuerzlinger e Danahy (2006), discriminam-se vários modos e estados de interação: o estado Design, oferecendo os modos Sketch, Draw e Modify; e o estado Action, oferecendo os modos Design e Observe. O botão direito do mouse muda de modo de interação.

Dorsey, Xu *et al.* (2007) ilustram a utilização do software elaborando vários objetos arquitetônicos a diferentes escalas. O mais representativo é uma casa unifamiliar, de geometria de base ortogonal e tetos a quatro águas, sem contexto urbano nem ambiental e com um terreno levemente sugerido (Figura 15).

Figura 15: Sequência de sketches que se sobrepõem em Mental Canvas para descrever uma casa.



Fonte: Dorsey, Xu *et al.* (2007).

O sistema de construção do Mental Canvas gera um sketch 3D a partir da sobreposição de sketches planos em diferentes orientações, denominados *canvas*. O processo começa pelo

desenho de um sketch sobre um *canvas*, no qual se representa uma edificação, seja em projeção paralela ou em perspectiva. Em outra orientação, se cria um outro *canvas*, no qual se desenha a mesma edificação sob um novo ponto de vista, no qual é possível representar outros elementos da composição. Por opção do usuário, é possível criar uma composição de *canvas* que abrange todo o contorno de uma edificação, a partir do qual a navegação simula a percepção de um modelo 3D. O processo parte do sketch 2D e, além da orientação dos *canvas*, não há modelagem 3D.

2.2 MODOS DE ENTRADA

Na utilização dos softwares, a primeira funcionalidade que define as características de interação humano-computador é o modo de entrada, instrumentado pelo dispositivo de entrada suportado. Excluindo o suporte ao uso de mouse e teclado, próprios da computação *desktop*, na Tabela 2 são comparadas as possibilidades oferecidas por estes softwares para a utilização dos modos de interação mais relevantes nas etapas iniciais de projeto.

Tabela 2: Características de interação dos softwares estudados

Referência	Direcionado à arquitetura	Dependência de teclado	Entrada por caneta	Entrada <i>multi-touch</i>	Entrada multimodal
Company, Contero <i>et al.</i> , 2003			•		
Pranovich, 2004	•	•			
Juchmes, Leclerq e Azar, 2004	•		•		
Kallio, 2005	•	•	•		
Masry e Lipson, 2005			•		
Oh, Stuerzlinger e Danahy, 2006	•	•			
Yu e Zhang, 2007	•				
Dorsey, Xu <i>et al.</i> , 2007	•	•	•		
Elsen e Leclerq, 2008	•		•		
Naya, Contero <i>et al.</i> , 2008			•		
Lopes, Mendes <i>et al.</i> , 2011			•	•	•
Elsen, Demaret <i>et al.</i> , 2012	•		•		
Kang, Kim <i>et al.</i> , 2013				•	

Fonte: o autor.

É perceptível na Tabela 2 que as experiências de Pranovich (2004), Oh, Stuerzlinger e Danahy (2006), e Yu e Zhang (2007), mesmo sendo direcionadas para as etapas iniciais de projetos de arquitetura, não incluem a utilização dos dispositivos de entrada mais relevantes nesta etapa—canetas *stylus* e telas *multi-touch*, dependendo da utilização do mouse para traçar

sketches. Porém, Yu e Zhang (2007) unificaram todas as funcionalidades para poderem ser acionadas a partir de um clique do mouse, prevendo a futura adaptação ao toque da caneta *stylus*.

2.2.1 Entrada por caneta *stylus*

Os softwares de Company, Contero *et al.* (2003) são concebidos para o desenho à mão livre com caneta *stylus* em computador *tablet* (com problemas de usabilidade devido a limitações do hardware disponível naquele momento). Em termos de sistemas de entrada, o software de Naya, Contero *et al.* (2008) se baseia nos mesmos dispositivos que Company, Contero *et al.* (2003).

Com um sistema de entrada genérico, Kallio (2005) aproveita qualquer dispositivo de entrada que atinja os padrões do sistema operacional. O autor apresenta o software com a utilização de uma caneta *stylus*. O software de Dorsey, Xu *et al.* (2007), assim como em Kallio (2005), é pensado para o seu uso combinado com caneta e teclado. Ambos dependem das flechas do teclado para posicionar os planos de construção em 3D.

O software de Juchmes, Leclercq e Azar (2004) e o protótipo evoluído a partir deste, em Elsen, Demaret *et al.* (2012), foram concebidos para a utilização de canetas *stylus* como dispositivo de entrada primário. O software de Elsen e Leclercq (2008) é desenvolvido especialmente para a sua utilização em mesas de projeção (*whiteboards*), com entrada direta por meio de uma caneta eletrônica. A implementação de software apresentada por Lopes, Mendes *et al.* (2011) é adaptada para trabalhar num *tabletop* ótico no qual se desenha com uma caneta de luz infravermelha, em combinação com outro modo de entrada (para oferecer um sistema multimodal a ser descrito no tópico correspondente).

O software de Masry e Lipson (2005) é direcionado à utilização de caneta *stylus*. A sua interface gráfica de usuário depende da entrada por caneta. O aplicativo de Kang, Kim *et al.* (2013) foi desenvolvido para a sua utilização unicamente com os dedos. Não aborda a entrada por caneta.

2.2.2 Entrada *multi-touch*

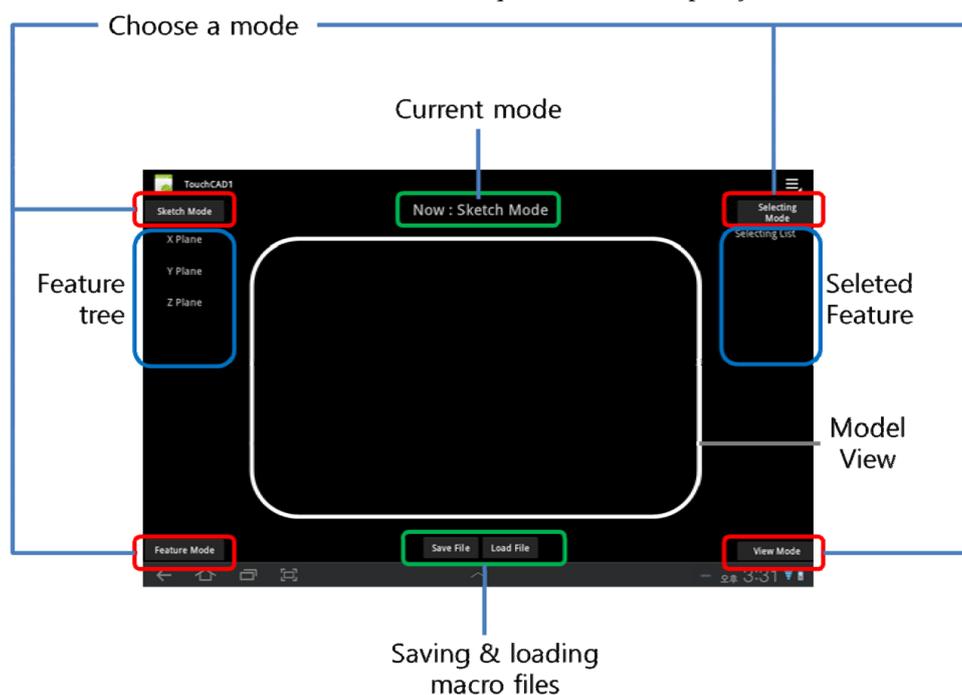
Tecnologias de entrada *multi-touch* existem desde 1982 (MEHTA, 1982), mas a primeira tecnologia acessível só foi comercializada a partir de 2007 (BUXTON, 2013), razão pela qual entende-se que os trabalhos publicados até essa data, delimitados nas possibilidades de hardwares comerciais, não tenham incluído o suporte desta funcionalidade de entrada direta.

Elsen e Leclercq (2008) e Naya, Contero *et al.* (2008), mesmo sendo contemporâneos à comercialização de tecnologias *multi-touch*, não abordam este tipo de entrada. Unicamente Lopes, Mendes *et al.* (2011) e Kang, Kim *et al.* (2013) aproveitam tecnologias *multi-touch* específicas para oferecer este tipo de entrada para controlar funcionalidades de visualização e edição de modelos 3D.

Em Kang, Kim *et al.* (2013), esta tecnologia é implementada a partir das funcionalidades de interação incluídas no sistema operacional Android, que conta com alto grau de adoção entre usuários finais. A interação foi otimizada para o uso *multi-touch* com uma só mão (considerando o uso em trabalhos de campo, onde a outra mão segura o computador *tablet*). A quantidade de funcionalidades de modelagem era muito grande para ser definida com gestos de uma só mão, podendo ser complexos de mais para garantir uma boa usabilidade. Por esta razão, Kang, Kim *et al.* (2013) introduziram quatro modos de operação baseados nas observações que fizeram das atividades de usuários de MCAD: o modo de sketch, o modo característica (*feature*), o modo de seleção, e o modo de mudança de vista.

Os diferentes modos de operação definem o efeito que os gestos *multi-touch* têm sobre o modelo e a sua visualização. A interface não é modificada na ativação de cada gesto, que é feita tocando em botões localizados nos quatro cantos da tela, um para cada modo operativo, segundo mostrado na Figura 16.

Figura 16: Design da interface gráfica de usuário do aplicativo de Kang, Kim *et al.* (2013), sinalizando os elementos de controle dos quatro modos de operação.



Fonte: Kang, Kim *et al.* (2013).

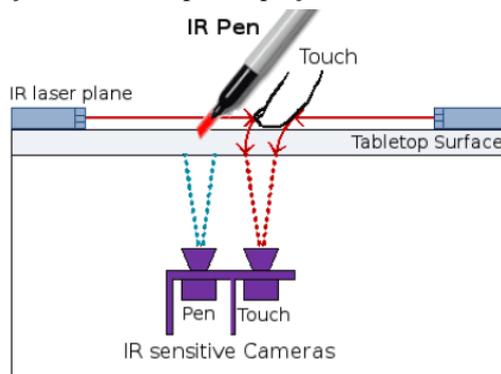
A localização estática dos elementos da interface provê consistência à interação, ao diminuir a carga cognitiva, reservando os quatro cantos da tela para os botões de seleção de modo de operação. A separação dos modos operativos acionados com estes botões não produz em si um problema de usabilidade. As ações de edição que o usuário executa nos modos diferentes ao modo de sketch são ações explícitas que acontecem num plano consciente. Ativar estas ações por meio de botões ou menus não quebra o processo cognitivo da projeção (COMPANY, CONTERO *et al.*, 2006). Por outro lado, o fato de mostrar em tempo real, numa janela lateral, a árvore de elementos e características que são gerados e modificados no processo de modelagem, pode fazer o usuário perder o foco e a continuidade de pensamento do seu processo de projeto (*op. cit.*).

Em Lopes, Mendes *et al.* (2011), o software funciona num *tabletop*. Recebe sinais de toques múltiplos quando o usuário coloca os seus dedos em contato com a superfície do *tabletop*, interrompendo um plano infravermelho formado por feixos que barrem a área de trabalho logo acima da superfície do *tabletop*. As diferentes combinações de toques definem um vocabulário de gestos que controlam algumas funções de navegação e manipulação do desenho.

2.2.3 Entrada multimodal

O trabalho de Lopes, Mendes *et al.* (2011) é o único que registra um aproveitamento de gestos bimanuais multimodais que combinam entrada de caneta e *multi-touch* simultaneamente para a edição de modelos 3D. Esta interação multimodal diminui a interrupção na continuidade do processo de pensamento do projetista na mudança de modos operativos ao minimizar protocolos de uso como o acionamento de botões ou menus, ainda presentes em softwares mais recentes, como o de Kang, Kim *et al.* (2013). Baseado na teoria da divisão assimétrica de tarefas bimanuais de Guiard (1987) e na implementação destes conceitos por Hinckley, Yatani *et al.* (2010a) (2010b), Lopes, Mendes *et al.* (2011) desenvolve um sistema de hardware com diferentes sensores para entrada com caneta ou com os dedos sobre uma tela *tabletop* ótica, ilustrado na Figura 17.

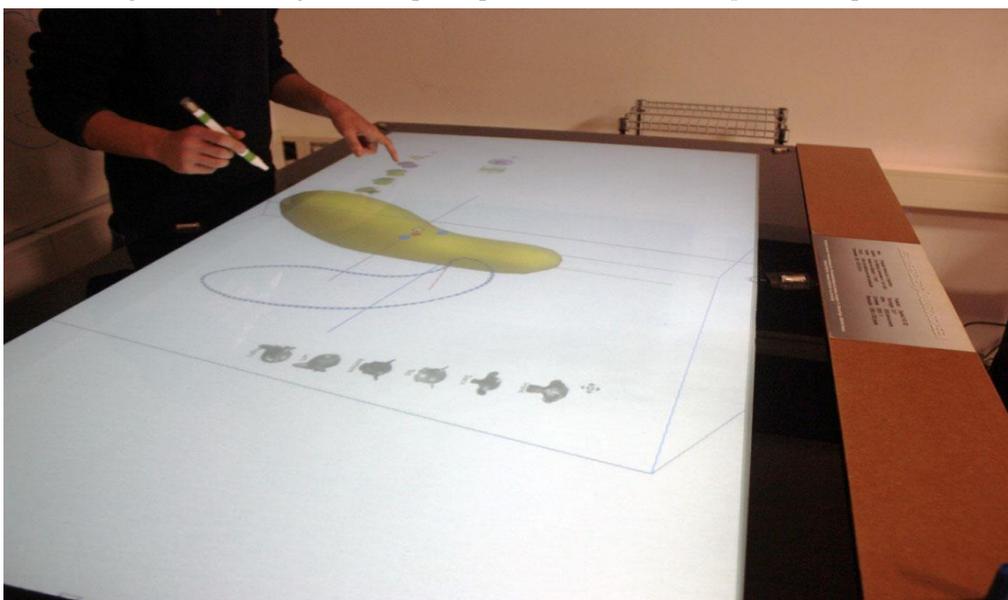
Figura 17: Esquema de adição de sensores para captação de entrada multimodal num *tabletop* ótico.



Fonte: Lopes, Mendes *et al.* (2011).

Na Figura 18 se registra a utilização desta configuração de hardware na execução do software ShapeShop (SCHMIDT, WYVILL *et al.*, 2005), atualizado por Lopes, Mendes *et al.* (2011) para aproveitar a entrada bimanual multimodal. Esta atualização de software o faz mais adequado para as etapas iniciais de projeto através de uma funcionalidade descrita na sequência.

Figura 18: Utilização do ShapeShop Multitouch no *tabletop* ótico adaptado.



Fonte: Lopes, Mendes *et al.* (2011).

2.3 FUNCIONALIDADES DE EDITORAÇÃO DE GEOMETRIA

As funcionalidades de desenho, modelagem e simulação oferecidas pelos softwares do estado da arte se apresentam comparados na Tabela 3.

Tabela 3: Funcionalidades para desenho, construção e visualização do modelo nos softwares estudados.

Referência	Aquisição de sketches	Reconhecimento	Reconstrução WIMP	Reconstrução automática	Reconstrução gestual	Simulação de desempenho
Company, Contero <i>et al.</i> , 2003	•	•		•		
Pranovich, 2004	•	•				
Juchmes, Leclerq e Azar, 2004	•	•		•		
Kallio, 2005	•					
Masry e Lipson, 2005	•	•		•		•
Oh, Stuerzlinger e Danahy, 2006	•	•	•			
Yu e Zhang, 2007	•	•	•			
Dorsey, Xu <i>et al.</i> , 2007	•					
Elsen e Leclerq, 2008	•	•				
Naya, Contero <i>et al.</i> , 2008	•	•			•	
Lopes, Mendes <i>et al.</i> , 2011	•	•			•	
Elsen, Demaret <i>et al.</i> , 2012	•	•		•		
Kang, Kim <i>et al.</i> , 2013	•	•			•	

Fonte: o autor.

Como pode ser percebido na Tabela 3, assim como estabelecido na fundamentação teórica³³, todas as experiências acadêmicas consideradas neste estudo oferecem funcionalidades de aquisição de sketches como base para o apoio computacional nas etapas iniciais de projeto. Porém, cada um aborda esta característica de maneira distinta e todos deixam de abordar outras funcionalidades.

Os softwares de Kallio (2005) e Dorsey, Xu *et al.* (2007) não geram sólidos, se focam no desenho de linhas em 3D. Isto lhes permite poupar a implementação de rotinas de reconhecimento, interpretação e reconstrução.

Oh, Stuerzlinger e Danahy (2006) e Yu e Zhang (2007) suportam o reconhecimento de formas a partir de traços adquiridos, mas logo abordam a reconstrução de uma maneira similar à modelagem tradicional, a partir de comandos acionados com recursos WIMP (grande quantidade de botões).

Em relação a métodos de captura da intenção do usuário, estudos precedentes sobre o estado da arte (COMPANY, CONTERO *et al.*, 2003) (OLSEN, SAMAVATI *et al.*, 2009) diferenciam duas aproximações ao problema de interação de softwares de modelagem baseada

³³ A aquisição de sketches é uma das funcionalidades que atende às atividades das etapas iniciais de projeto, com o potencial para aumentar o suporte de softwares CAD nestas etapas, e assim atingir às demandas da profissão. Veja-se no ponto 1.2.6 Interação multimodal relevante para o processo de projeto, página 38.

em sketches: o **método automático** e o **método gestual**³⁴. Estas duas aproximações diferenciam duas linhas de pesquisa originadas por estes acadêmicos: (i) na abordagem automatizada, produzindo o software REFER (COMPANY, CONTERO *et al.*, 2003); e (ii) na abordagem gestual, produzindo os softwares CIGRO, ParSketch e GEGROSS (CONTERO, NAYA *et al.*, 2003) (NAYA, CONESA *et al.*, 2003) (NAYA, CONTERO *et al.*, 2004) (NAYA, CONTERO *et al.*, 2008). Os softwares de Company, Contero *et al.* (2003) e de Masry e Lipson (2005) geram modelos 3D a partir de sketches em perspectiva por meio de funcionalidades de reconstrução automática explicadas na sequência.

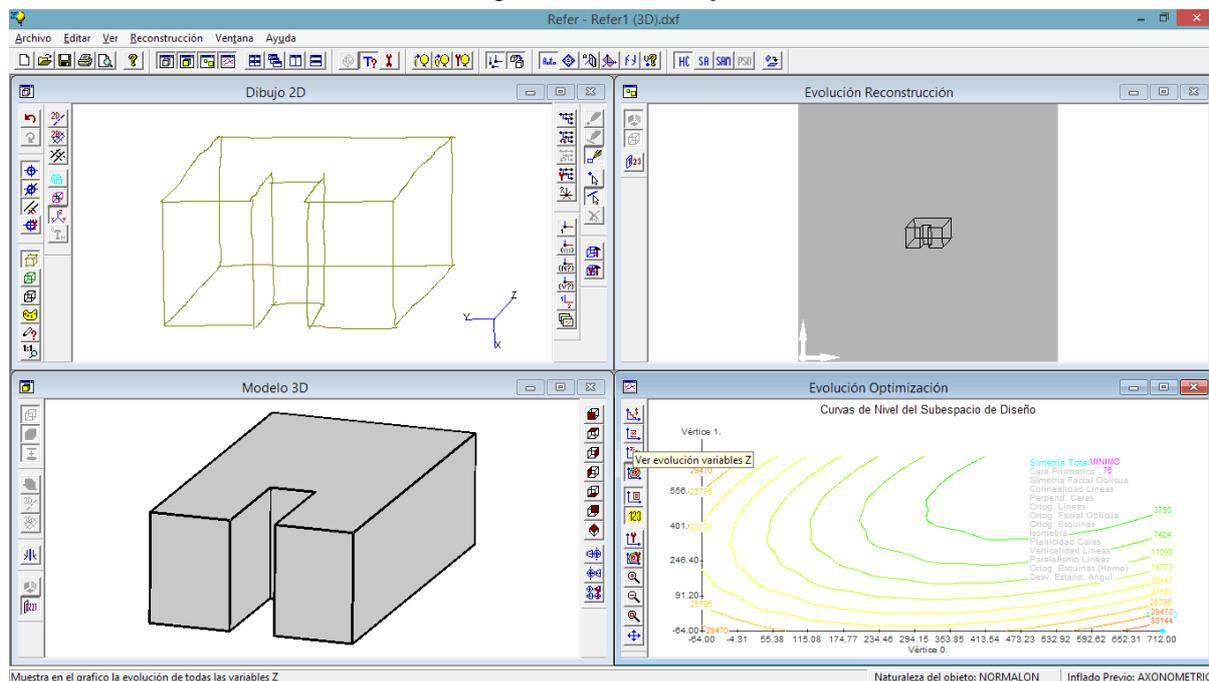
2.3.1 Reconstrução automática

Centra-se em algoritmos de interpretação da tridimensionalidade do sketch em perspectiva desenhado pelo usuário (COMPANY, CONTERO *et al.*, 2003). Este sketch é usado como recurso estático de entrada para a reconstrução, numa postura WYSIWYG³⁵, com o intuito de oferecer uma utilização intuitiva desde o ponto de vista das instruções de desenho. O projetista deve completar o desenho que represente a geometria 3D de um objeto em perspectiva. Na prática, o problema está na mudança de atividade (selecionando botões e preenchendo valores numéricos) para ativar a reconstrução, na qual o software toma em consideração todos os traços desenhados como dados de entrada para a função de interpretação. Na Figura 19, se apresenta este processo com um exemplo desenhado em REFER, selecionando como método de reconstrução um algoritmo de otimização.

³⁴ Na literatura (COMPANY, PIQUER e CONTERO, 2004), estes métodos são mencionados como “método automático e método interativo”. Os mesmos autores fazem uma revisão desses termos em trabalhos posteriores (COMPANY, CONTERO *et al.*, 2006), reconfirmados pelo estudo de Olsen, Samavati *et al.* (2009) como “método reconstrutivo e método gestual”. Esta postura não é utilizada aqui, já que assume, erradamente, que a reconstrução é um processo inerentemente automático, quando este pode ser também interativo através de comandos WIMP ou gestos.

³⁵ WYSIWYG: *what you see is what you get*. Veja-se o ponto 1.2.2 Evolução da interação por teclado e mouse, da fundamentação teórica, página 26.

Figura 19: Interface do REFER na reconstrução automática de um sketch em perspectiva cavaleira por meio de algoritmos de otimização.



Fonte: sketch nosso, na utilização do software de Company, Contero *et al.* (2003).

A reconstrução por otimização é suportada em REFER pela implementação de vários algoritmos genéricos, dentre os quais se oferece o Hill Climbing como a opção por omissão (por *default*)³⁶. Este algoritmo permite encontrar uma boa solução dentre muitas soluções possíveis, dadas as variáveis de um problema, avaliadas por uma função de aptidão (SKALAK, 1994). Neste caso, o algoritmo utiliza como variáveis as possíveis combinações de coordenadas 3D dos vértices reconhecidos no sketch, que são mapeadas no espaço bidimensional dos possíveis planos de perspectiva deste sketch. Na Figura 19, a janela inferior direita mostra a representação de um gráfico com a definição numérica do espaço de soluções, sendo definido pelo algoritmo Hill Climbing. O usuário pode modificar os critérios de busca de soluções aptas a partir de uma janela de configurações, que é mostrada na Figura 20 com os valores por *default*.

³⁶ Na versão de REFER experimentada no presente estudo (versão 4.0 de 10 out. 2003), os métodos de otimização suportados são Hill Climbing (escalada), Simulated Annealing (recozimento simulado) e Simulated Annealing multicritério. Uma descrição sobre os parâmetros destes algoritmos pode ser acessada através de botões, descritos no decorrer deste texto.

Figura 20: Janela de configuração dos parâmetros da otimização Hill Climbing para reconstrução automática em REFER.



Fonte: captura nossa, na utilização do software de Company, Contero *et al.* (2003).

Valores *por default*, ou por omissão são os valores que vêm predefinidos no software e podem ser modificados pelo usuário. Esta janela de configuração (Figura 20), assim como todas as do software, oferece a opção de voltar a estes valores por meio do botão “*Restaurar valores por defecto*”. O botão “*Detalles*” leva à janela apresentada na Figura 21, onde são informados os valores admissíveis, e são explicadas as consequências deles na execução desta função.

Figura 21: Janela de detalhes de configuração da otimização Hill Climbing para reconstrução 3D no REFER.

PARAMETROS	FUNCION	RANGO				VALORES RECOMENDADOS PARA		
		Minimo		Maximo		Velocidad	Precisión	Funcional
		Valor	Observaciones	Valor	Observaciones			
Longitud de paso inicial	Escapar de optimos locales	1	Convergencia lenta (Da varios pasos seguidos en la misma variable)	25 (10)	Se escapa de optimos locales (pasos demasiado grandes). Peligro mayor en piezas complejas	10	2	2.5
Reducción de paso	Complementa al parámetros "longitud de paso"	10	Los incrementos decrecen antes de tiempo. La convergencia se desacelera	50	Aborta porque no consigue acercarse bastante al optimo (oscila alrededor del optimo)	20	40	50
Longitud de paso final	Rapidez/precisión	2	Las partes pequeñas no se reconstruyen (salvo que también disminuya el parámetros "longitud de paso" del mínimo)	10	La variación del incremento de variables es inapreciable. Mejor aumentar el parámetro "Pasos por escalón"	0.5	0.06	0.15
Pasos por escalon	Complementa al parámetro "numero de pasos"	50	Aborta antes de terminar cada búsqueda lineal (puede disminuir cuando al parámetro "longitud de paso" aumenta a 25 y el parámetros "pasos por escalon" disminuye a 20)	500	Tarda mucho tiempo cuando los últimos incrementos son pequeños (parámetro "Reducción de paso" pequeño y parámetro "numero de pasos" grande)	100	500	100
Disminución de coste		0	Influye en la búsqueda lineal	1	Las mejoras en cada paso son siempre pequeñas. Nunca se pueden exigir mejoras grandes en un solo paso	0	0	0

Botón: Aplicar

Fonte: captura nossa, na utilização do software de Company, Contero *et al.* (2003).

Ainda considerando o aproveitamento da tecnologia de entrada por caneta no desenho do sketch, a interação neste software é altamente dependente de elementos WIMP e os seus

estilos de interação clássicos, através de uma interface “cifrada” por protocolos de configuração e terminologia das ciências da computação, alheios ao conhecimento a ser contemplado nas disciplinas de desenho onde o software foi testado. Mesmo assim, reportaram altos níveis de aceitação na população testada por eles (COMPANY, CONTERO *et al.*, 2003).

2.3.2 Reconstrução gestual

Esta abordagem se baseia no uso de gestos de desenho que o usuário deve executar em combinação com o desenho dos elementos em processo de modelagem. Cada gesto substitui uma funcionalidade CAD que, de outra maneira, seria ativada por meio de botões ou comandos de texto, típicos da maneira de trabalhar em softwares CAD. Na literatura estudada, estes gestos referem-se a desenhos, ou símbolos que o usuário traça com a caneta ou com o toque dos dedos, diretamente no desenho³⁷. Uma vez que estes gestos são reconhecidos, a função é desempenhada pelo software, substituindo o desenho do gesto pelo produto desta ação numa postura WYSIWYM³⁸.

Mesmo na abordagem automática, o sistema de interação do CIGRO (CONTERO, NAYA *et al.*, 2003) já incluía alguns gestos, utilizando a biblioteca de programação CALI (FONSECA e JORGE, 2001), que oferece funções de interpretação caligráfica assim como em REFER (COMPANY, CONTERO *et al.*, 2003). Entre os gestos encontra-se o de apagar linhas (“Borrar”), definido como um rabisco em ziguezague sobre a linha a ser apagada (*Cross-out*) (Figura 22).

Figura 22: Janela de ajuda sobre os gestos reconhecidos em REFER e CIGRO.



Fonte: Captura nossa, na utilização do software de Company, Contero *et al.* (2003).

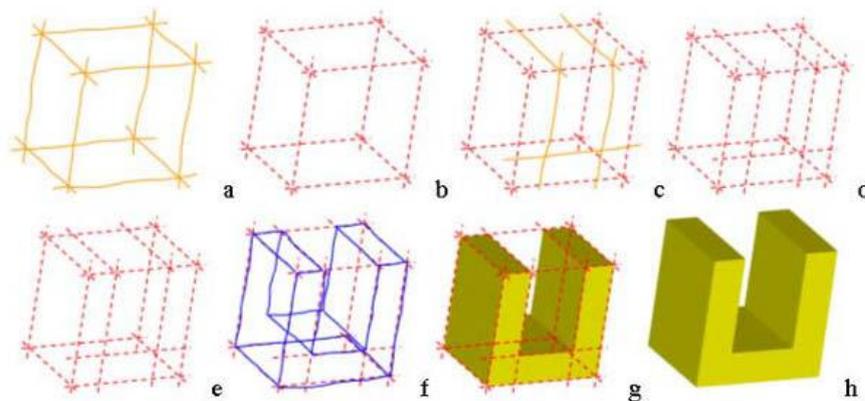
Diferente do REFER, o CIGRO não explora funcionalidades automáticas de otimização, e sim uma abordagem totalmente gestual. O modelo 3D é conseguido a partir do

³⁷ Sistemas de interação que utilizam gestos feitos literalmente gesticulando com as mãos, como em Krueger (1983), são desconsiderados neste estudo por precisar de outros dispositivos periféricos, como câmeras e outros recursos de software, como a captura e reconhecimento de padrões de vídeo, conjunto de recursos que ainda não atingiram o nível de adoção e onipresença das tecnologias de *tablet* e caneta.

³⁸ WYSIWYM: *what you see is what you meant*. Veja-se o ponto 1.2.2, página 24.

reconhecimento e interpretação em tempo real das linhas do sketch, desenhadas sobre planos de construção que são previamente definidos por linhas auxiliares. O usuário instrui o software como interpretar os traços por meio da pressão exercida com a caneta sobre a tela. Traços fracos definem linhas auxiliares, e traços fortes linhas do modelo. Um exemplo de processo de modelagem se ilustra na Figura 23.

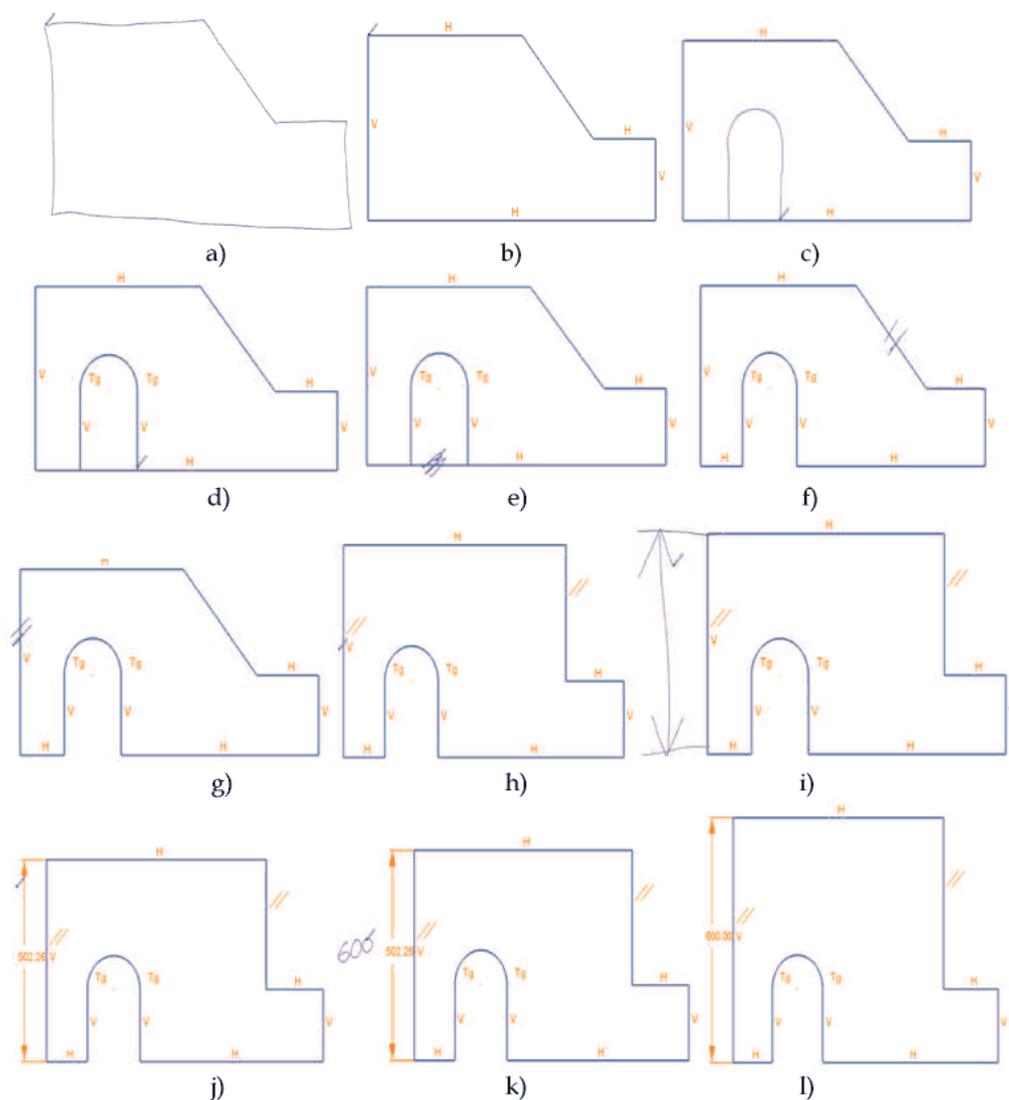
Figura 23: Processo de modelagem em CIGRO a partir do reconhecimento e interpretação em tempo real.



Fonte: Naya, Conesa *et al.* (2003).

Avanços deste grupo de pesquisadores produziram depois o sistema denominado ParSketch, que implementava uma maior quantidade de símbolos CALI para o reconhecimento e interpretação de sketches prescritivos, suportando a definição de restrições geométricas e dimensionais próprias do desenho paramétrico. O sistema suportava restrições geométricas como paralelismo e perpendicularidade, e restrições dimensionais, no referido a comprimentos dos elementos, definidos por desenhos de cotas. Originalmente, funcionava só em 2D, mas oferecia uma maior liberdade geométrica com respeito ao CIGRO ao suportar o reconhecimento, não só de linhas, como também de círculos e arcos (NAYA, CONESA *et al.*, 2003). A Figura 24 apresenta uma sequência de desenho de uma peça representada, em corte, por um contorno composto de segmentos de linhas e arcos (NAYA, CONTERO *et al.*, 2008).

Figura 24: Sequência de desenho paramétrico por restrições em ParSketch.



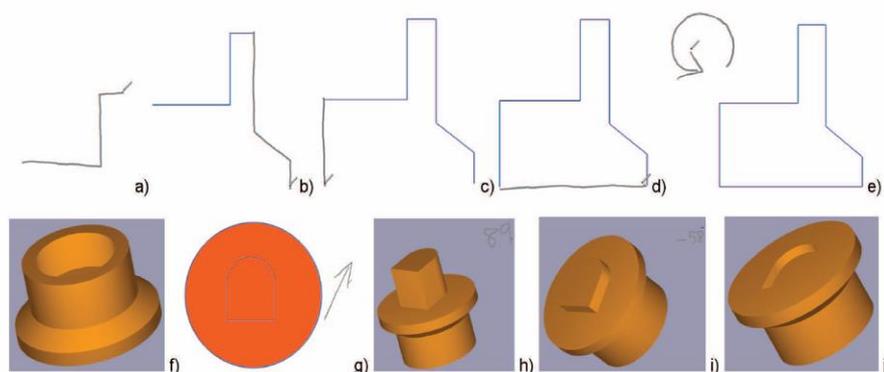
Fonte: Naya, Contero *et al.* (2008).

Neste exemplo (Figura 24), os traços em sketch (a) são reconhecidos no momento em que o usuário levanta a caneta da tela, identificando segmentos de linhas horizontais e verticais (b). Após o desenho de uma linha poligonal que contém um arco (c), o software reconhece os segmentos, interpretando a intenção de forçar a tangência entre as linhas e o arco (d), e com o gesto *cross-out*, apaga-se o segmento sobrando (e). Depois, o usuário decide mudar o segmento inclinado (f), fazendo com que seja paralelo com o maior segmento vertical (g, h). Finalmente, controla a altura da peça, definindo uma linha de cota (i, j) e substituindo o valor da altura por um valor arbitrado (k, l).

ParSketch é posteriormente utilizado como módulo de desenho para o sistema de modelagem em 3D, baseada em sketch GEGROSS, introduzindo uma nova categoria de símbolos gestuais: os gestos de modelagem. Está registrada na literatura a implementação de

três gestos de modelagem: estruair, revolucionar em sentido horário e revolucionar em sentido anti-horário (Figura 25).

Figura 25: Sequência de modelagem em GEGROSS. Exemplo que combina gestos de revolução e extrusão.



Fonte: Naya, Contero *et al.* (2008).

Para as funcionalidades de desenho, assim como as de seleção e manipulação, o software de Kang, Kim *et al.* (2013) utiliza a abordagem gestual, repensada a partir da implementação feita por Kim e Kim (2006) da biblioteca CALI. Assim, redefinem todo o vocabulário de traços gestual deles para oferecer uma interação com o toque dos dedos, para desenhar linhas e curvas e controlar ações de seleção e manipulação. Além do desenho 2D, definiram um vocabulário de gestos de modelagem 3D para gerar superfícies e sólidos, como extrusão, revolução, transição (*loft*), entre outros (*ibid.*).

Para poder oferecer um sistema de software suficientemente leve para dispositivos móveis, Kang, Kim *et al.* (2013) estudaram a frequência de uso das ações de modelagem MCAD e selecionaram só as mais frequentes para garantir que, mesmo com uma funcionalidade reduzida, a implementação deles tivesse a maior compatibilidade possível³⁹. Ainda, desta seleção, substituíram algumas funcionalidades não tão frequentes para conseguir os mesmos resultados com a combinação de outras ações mais frequentes.

2.3.3 Sketches em 2D ou 3D

Não são todos os softwares estudados que permitem a modelagem em 3D. Pranovich (2004) e Elsen e Leclercq (2008) desenvolvem sistemas de desenho de plantas arquitetônicas, implementando conceitos de composição arquitetônica como grelhas e outras unidades gráficas (PRANOVICH, 2004), dando suporte à definição de informações semânticas e dados

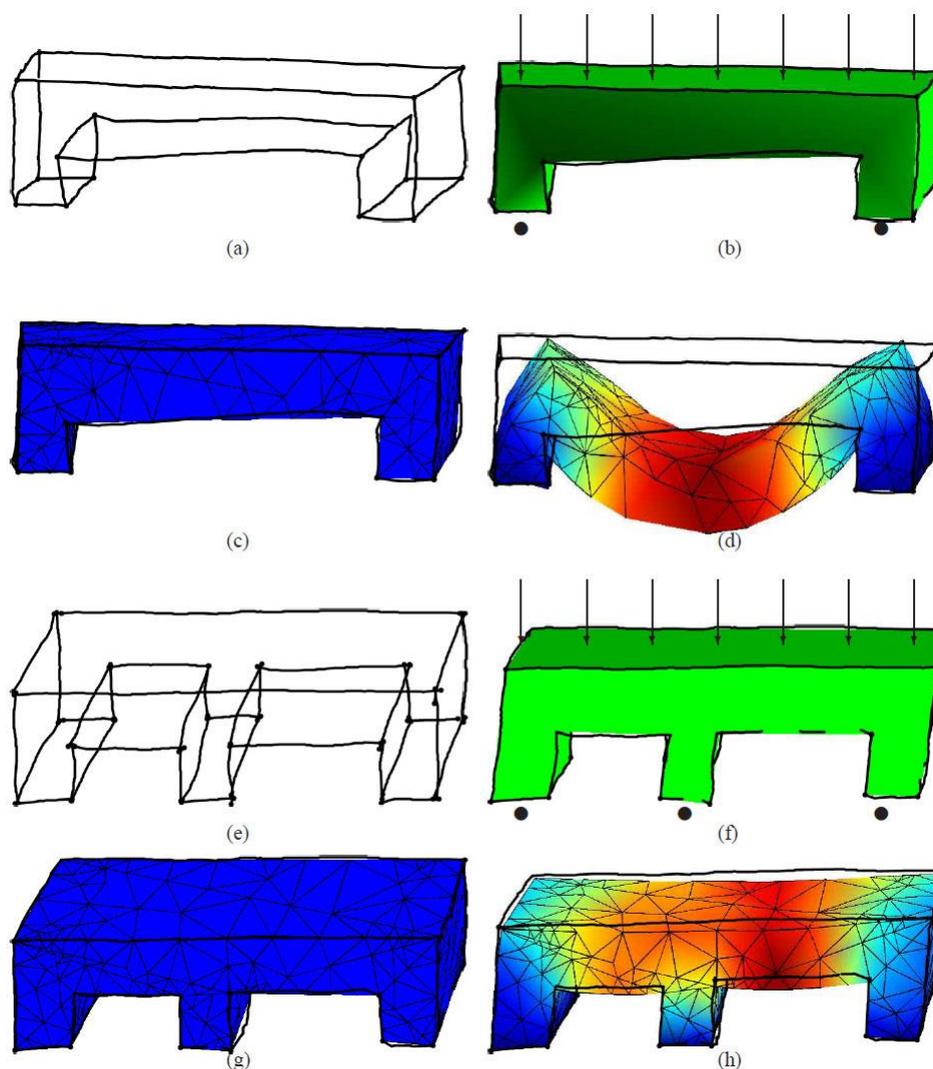
³⁹ Através de um estudo com usuários, foi determinado que as ações de modelagem mais comuns na engenharia são: *Extrusion*, *Cut extrusion*, *Chanfer*, *Linear pattern*, *Cut Revolve*, *Datum plane*, *Revolve*, *Circular pattern*, *Fillet* e *Loft* (KANG, KIM *et al.*, 2013).

contextuais, num ambiente bidimensional (ELSEN e LECLERCQ, 2008). Esta funcionalidade é relevante para as tomadas de decisão sobre o programa arquitetônico (uso e requerimentos de área dos espaços) e o *layout*, importantes nas etapas iniciais de projeto arquitetônico (*ibid.*). A utilização de grelhas também é suportada em Kallio (2005) para definir orientação e escala de desenhos definidos em 2D, mas ordenadas no espaço 3D. Os traços introduzidos pelo usuário são projetados no plano da grelha definida. Porém, não suporta transformações ou alterações após a criação dos traços, e é necessária uma grande quantidade de tempo para elaborar um sketch (DORSEY, XU *et al.*, 2007). Dorsey, Xu *et al.* (*ibid.*) utilizam a mesma aproximação de planos posicionados no espaço tridimensional, com a adição de funcionalidades de edição e gestão dos atributos dos traços, independentemente do plano em que foram criados, oferecendo ao usuário a oportunidade de aclarar situações ambíguas (*ibid.*).

2.3.4 Simulação de desempenho

Só Masry e Lipson (2005) implementam funcionalidades de simulação de desempenho em combinação com um sistema de modelagem baseado em sketch. O fator de desempenho explorado, neste caso, é o desempenho estrutural. Na Figura 26, está ilustrado um exemplo de utilização, no qual o projetista gera um pórtico estrutural genérico, especifica forças e, em seguida, o software executa uma análise de esforços pelo método de elementos finitos (FEA) e apresenta a simulação de desempenho com os resultados da análise, neste caso, de deformações, tanto numa variação cromática como num deslocamento dos vértices do modelo amostrado. Este feedback ajuda o projetista a visualizar um problema de desempenho na sua ideia de projeto, no momento imediatamente posterior ao seu esboço. Assim, numa segunda iteração, adiciona um apoio vertical ao vão do pórtico previamente desenhado e, após redefinir os parâmetros de forças, confere a melhoria no desempenho em questão de segundos.

Figura 26: Exemplo de sessão de projeção por reconstrução automática de sketches e simulação de desempenho estrutural em duas iterações.



Fonte: Masry e Lipson (2005).

2.3.5 Computação e interoperabilidade do modelo

Com respeito à computação do modelo com a qual trabalham estes softwares, na Tabela 4 se estudam as características: paradigma de modelagem, diversidade geométrica e interoperabilidade com softwares de apoio nas etapas avançadas de projeto, como dados para avaliar a possibilidade de transição entre etapas.

Tabela 4: Características de computação dos softwares experimentais estudados.

Referência	Paradigma de modelagem	Diversidade geométrica	Interoperabilidade CAD
Company, Contero <i>et al.</i> , 2003	Modelagem por reconstrução	Sólidos de faces planas	Alta (exportando DXF)
Pranovich, 2004	Desenho arquitetônico vetorial	Polígonos 2D	Nenhuma
Juchmes, Leclercq e Azar, 2004	Desenho de sketches	Sólidos por extrusão de polígonos	N/A
Kallio, 2005	Desenho de sketches	Linhas 3D	Alta (exportando VRML)
Masry e Lipson, 2005	Modelagem por reconstrução	Sólidos irregulares triangulados	N/A
Oh, Stuerzlinger e Danahy, 2006	Modelagem por extrusão	Sólidos por extrusão de polígonos	Alta (exportando 3ds)
Yu e Zhang, 2007	Modelagem por extrusão	Sólidos por extrusão de polígonos	Nenhuma
Dorsey, Xu <i>et al.</i> , 2007	Desenho de sketches	Linhas 3D	Média (exportando desenhos 2D)
Elsen e Leclercq, 2008	Desenho de sketches	Linhas e polígonos 2D	Nenhuma
Naya, Contero <i>et al.</i> , 2008	CAD baseado em sketch	Sólidos de faces planas e em arco	Alta (exportando DXF)
Lopes, Mendes <i>et al.</i> , 2011	Modelagem por reconstrução	Isosuperfícies (Blobs)	Alta
Elsen, Demaret <i>et al.</i> , 2012	Desenho de sketches	Sólidos de faces planas e em arco	N/A
Kang, Kim <i>et al.</i> , 2013	MCAD macro-paramétrico (MPA)	Sólidos e superfícies de forma livre	Média (plug-in de macros)

Fonte: o autor.

Como pode ser conferido na Tabela 4, existe uma menor diferenciação e uma maior sobreposição destas características. Company, Contero *et al.* (2003) e Masry e Lipson (2005) utilizam a reconstrução 3D de sketches desenhados em perspectiva para gerar modelos sólidos de faces planas. Oh, Stuerzlinger e Danahy (2006) e Yu e Zhang (2007) se baseiam na modelagem por extrusão para gerar modelos 3D, apoiados no fato de ser o tipo de modelagem que melhor acompanha a lógica de projeção de espaços arquitetônicos, tradicionalmente de forma prismática, que otimizam os materiais de construção restringidos pela força de gravidade e pelos padrões das indústrias de revestimentos e de mobiliário (YU e ZHANG, 2007). Juchmes, Leclercq e Azar (2004) também se fundamentam nesta lógica de projeção, abordando-a desde a técnica de representação mais aproximada ao desenho de sketches arquitetônicos. Esta técnica de representação é utilizada por Kallio (2005) de maneira diferente. O 3D6B Editor é muito flexível na geração de sketches em 3D e oferece interoperabilidade ao exportar em formato VRML, mas não permite a definição explícita de sólidos, nem a geração direta destes em outros softwares, já que a diversidade geométrica com a qual trabalha se limita a linhas 3D. Não há possibilidade de criar outros elementos geométricos, nem de definir informações de projeto arquitetônico, além da espessura e cor das linhas. Em contraposição, o protótipo de Yu e Zhang (2007) aborda tanto a naturalidade de elaboração de sketches

arquitetônicos, como a geração de sólidos, que poderiam ser aproveitáveis nas etapas seguintes do processo de projeto se o software suportasse a exportação do modelo, o que não é possível.

Contero, Naya *et al.* (2003) adotaram inicialmente, no desenvolvimento do CIGRO, o método automático, podendo reconstruir sketches desenhados em perspectiva isométrica, assim como o REFER (COMPANY, CONTERO *et al.*, 2003). O reconhecimento (por eles chamado de “reconstrução 2D”) e a reconstrução (3D) implementam algoritmos recursivos de otimização que só atingem um nível de eficácia favorável quando o usuário especifica uma configuração de parâmetros de interpretação rígida, limitando a amplitude potencial de ação em termos de variedade de geometrias e ângulos de perspectiva⁴⁰.

Posteriormente, mudaram a abordagem baseada no método gestual (NAYA, CONTERO *et al.*, 2008), o que trouxe ganhos em eficácia e usabilidade, além de permitir-lhes introduzir novas funcionalidades, como a definição de restrições paramétricas (*ibid.*).

O motor geométrico lê e escreve os dados do modelo 3D em formato DXF, um formato padrão de transferência de arquivos de CAD. Isto permite uma alta interoperabilidade com todos os softwares de CAD comerciais.

Devido aos softwares terem sido construídos em ambiente Windows XP, para ser executado na primeira geração comercial de computadores *tablet* que usa este sistema operacional (Toshiba Tecra M4), o sistema de interação não suporta *multi-touch*. As funcionalidades de navegação e edição são acessadas através de uma grande quantidade de botões que ocupam área da tela.

O software protótipo de aplicativo MCAD baseado em sketch, apresentado por Kang, Kim *et al.* (2013), é pensado para engenheiros utilizarem nas etapas iniciais de projeto, criando modelos por meio de interação com tela *multi-touch*. Um dos principais objetivos é o de prover um meio de iniciar projetos fora do escritório, utilizando pequenos computadores *tablet* e outros dispositivos móveis, e depois poder continuar elaborando-os no escritório, utilizando pacotes de software MCAD comerciais em computadores *desktop*, sem ter que investir tempo e perder informação em processos de exportação e importação de modelos de diferentes formatos (*ibid.*). Para isto, o sistema de software lê e escreve arquivos de texto que contêm listas de instruções de comandos textuais ou *macros*, utilizados nos principais softwares de MCAD comerciais.

⁴⁰ Descreve-se a abrangência de possíveis geometrias a serem elaboradas com os softwares REFER e CIGRO como “normalones” e “quase-normalones”, ou seja, sólidos de faces ortogonais ou quase ortogonais. Os autores (COMPANY, CONTERO *et al.*, 2003) (NAYA, CONESA *et al.*, 2003) identificam esta especificidade como uma limitação a ser superada em futuros trabalhos. A superação que os autores conseguiram desta limitação vem em GEGROSS, através da mudança fundamental do método de reconstrução como descrito na página 57 do ponto 2.3.2 Reconstrução gestual.

Esta aproximação à modelagem é autodenominada macro-paramétrica (*macro-parametric approach* - MPA) (KANG, KIM *et al.*, 2013), para a qual o aplicativo armazena os dados geométricos temporalmente, só registrando os macros num arquivo de texto ASCII (*ibid.*). Para garantir a interoperabilidade com os diferentes softwares MCAD, foram desenvolvidos *plugins* de tradução do macro do seu formato neutral à linguagem de macro destes softwares, para que possa ser carregado e executado, e assim regenerar o modelo originalmente criado no computador *tablet*.

Devido a o software não ter sido pensado para entrada por caneta, e sim com os dedos, a precisão é reduzida, sendo ainda apropriada para as atividades das etapas iniciais de projeto. Esta redução também foi considerada nas funcionalidades de manipulação, que não suportam a edição nem seleção de vértices (*op. cit.*).

2.4 POTENCIAL DE REAPROVEITAMENTO E INTEGRAÇÃO DAS FUNCIONALIDADES

Para entender o potencial reaproveitamento destas funcionalidades de interação, modelagem e visualização, e demais características dos softwares, é necessário analisar o contexto computacional no qual estes são desenvolvidos. Uma das características que define a possibilidade de aproveitamento destas funcionalidades de software com determinados hardwares, é a linguagem de programação na qual são desenvolvidas. Determinadas linguagens e implementações são compatíveis com alguns sistemas operacionais, e não com outros. Por exemplo, as versões modernas de Visual Basic e C# fazem parte da plataforma Microsoft .NET, desenvolvida para ser empregada em sistemas operacionais Microsoft Windows, não suportando a sua execução em outros sistemas operacionais⁴¹ (CZELUSNIAK, BINHARA *et al.*, 2009). Para um programa de computador ser utilizado pelos usuários, este deve ser compilado pelo desenvolvedor como um aplicativo executável. Esta compilação deve ser feita num computador que funcione com o sistema operacional para o qual se destina este aplicativo. O código fonte de softwares desenvolvidos nestas linguagens não pode ser compilado em computadores que utilizam outros sistemas operacionais, como o Apple iOS, com o qual trabalham os populares computadores *tablet* iPad. Isto também é certo para softwares desenvolvidos em C++, se fizerem uso da plataforma .NET. Uma relação de incompatibilidade similar, mas na direção oposta, existe com linguagens como Objective-C ou Macintosh

⁴¹ A plataforma .NET se apresenta como multi-plataforma, já que é compatível com vários sistemas operacionais Microsoft, como Windows e Windows Phone. MICROSOFT CORPORATION. Multiple Platform Support. In: **Microsoft .NET Framework**. Disponível em: <<http://www.microsoft.com/net/multiple-platform-support>>. Acesso em: 20 jul 2014.

Common Lisp (BROOKS e ROSENBERG, 1995), que trabalham nos sistemas operacionais da Apple e são incompatíveis com os sistemas Windows.

Alternativamente, linguagens como Oracle Java são multi-plataforma (WANG e WANG, 2001), significando que os códigos fonte escritos nestas linguagens são compiláveis em diferentes sistemas operacionais e que alguns aplicativos compilados são executáveis em mais de um sistema operacional.

Sem a pretensão de discorrer sobre as razões corporativas desta incompatibilidade, a discussão sobre o funcionamento de softwares em determinados computadores faz parte importante do contexto tecnológico das experiências acadêmicas analisadas. A aceitação que apresenta a tecnologia *multi-touch* se deve a ampla adoção de dispositivos como o iPad por usuários finais a partir de 2010 (INGRAHAM, 2013). Mas grande parte do estado da arte aqui encontrado trabalha sobre sistemas operacionais Windows, que suportam nativamente entrada direta por caneta (na tela do computador) desde 2002 (CONTERO, NAYA *et al.*, 2003), e por *multi-touch* desde 2009 (KANG, KIM *et al.*, 2013). Na evolução destas tecnologias, temos visto avanços significativos desde 1963 (SUTHERLAND, 1963), mas para atingir uma ampla adoção destas tecnologias por parte dos arquitetos, é necessário centrar-se nas tecnologias disponíveis para usuários finais. Esta disponibilidade só acontece a partir da década de 2000 (BUXTON, 2008). Na Tabela 5, apresenta-se a comparação de linguagens empregadas e a compatibilidade de sistemas operacionais oferecida pela literatura selecionada.

Percebe-se, na Tabela 5, que a maioria dos softwares experimentais é desenvolvida em C++ e oferece compatibilidade só com computadores que operam Windows. Comparando com o suporte à entrada por caneta apresentado na Tabela 2, pode-se ver que a maioria dos softwares compatíveis com Windows suporta entrada por caneta, mas não *multi-touch*. Isto pela existência de versões Tablet-PC Edition de Windows XP para as primeiras gerações de computadores *tablets* para usuários finais utilizadas nestas experiências acadêmicas. Estas tecnologias (a tela e o sistema operacional) são anteriores à comercialização dos primeiros equipamentos *multi-touch*. Percebe-se também que os softwares operados exclusivamente em Macintosh são desenvolvidos em Macintosh Common Lisp.

Tabela 5: Linguagens de programação empregadas no estado da arte e compatibilidade oferecida.

Referência	Linguagem de programação	Compat. Windows	Compat. Mac OS	Compat. iOS	Compat. Android
Company, Contero <i>et al.</i> , 2003	C++	•			
Pranovich, 2004	C++	•			
Juchmes, Leclerq e Azar, 2004	Macintosh Common Lisp		•		
Kallio, 2005	Java	•	•		
Masry e Lipson, 2005	C++	•			
Oh, Stuerzlinger e Danahy, 2006	N/A	•			
Yu e Zhang, 2007	C++	•			
Dorsey, Xu <i>et al.</i> , 2007	C++	•			
Elsen e Leclerq, 2008	Macintosh Common Lisp		•		
Naya, Contero <i>et al.</i> , 2008	C++	•			
Lopes, Mendes <i>et al.</i> , 2011	C++	•			
Elsen, Demaret <i>et al.</i> , 2012	Macintosh Common Lisp		•		
Kang, Kim <i>et al.</i> , 2013	Java				•

Fonte: o autor.

Company, Contero *et al.* (2003) desenvolveram, em C++, no software REFER, um sistema de modelagem baseada em sketch para computadores *tablet* com entrada por caneta *stylus*. O software foi testado pelos autores num Toshiba Tecra M4, operando Windows XP Tablet-PC Edition, assim como, posteriormente, em Naya, Contero *et al.* (2008) (Figura 9, página 37). O potencial de aproveitamento dos algoritmos implementados nestes softwares é amplo no referido a sistemas operacionais modernos da família Windows, como o Windows 8 e o Windows Phone, que hoje em dia fazem uso extensivo de métodos de entrada direta como o *multi-touch*. Porém, a potencial integração com sistemas operacionais com maior crescimento em relação a estes métodos de entrada direta, como o iOS e o Android, não é clara.

Percebe-se, também, que só os softwares de Kallio (2005) e de Kang, Kim *et al.* (2013) são desenvolvidos em linguagens multiplataforma, especificamente Java. Porém, a implementação de interação por entrada direta em Kang, Kim *et al.* (2013) é específica para sistemas Android, em vez de tomar uma aproximação genérica, como a de Kallio (2005). Em última instância, uma integração destes dois softwares seria inviável, devido a estarem baseados em diferentes paradigmas de modelagem, como foi verificado na Tabela 4.

Muitos dos softwares experimentais descritos abordam, dentre as suas funcionalidades, algum suporte apropriado às atividades das etapas iniciais de projeto arquitetônico. Porém, devido a questões fundamentais, como o paradigma de base, ou a questões práticas, como a compatibilidade de hardware, não há uma maneira direta de integração dos programas de

computador apresentados. Seria necessário o desenvolvimento de um novo software que implemente estas funcionalidades de maneira integrada. Para poder desenvolver um software como este é necessário, primeiramente, extrair os princípios de funcionalidade que há por trás de cada software estudado, desde a perspectiva do pesquisador em interação humano-computador, considerando as atividades projetuais do arquiteto.

3 FUNCIONALIDADES NECESSÁRIAS PARA APOIAR AS ETAPAS INICIAIS DE PROJETO ARQUITETÔNICO

Tendo em consideração a crítica ao estado da arte apresentada, definem-se as funcionalidades que um software precisa ter para o apoio às etapas iniciais de projeto arquitetônico. Este software deve aproveitar as tecnologias que, como exposto na fundamentação teórica, estão disponíveis e são as mais relevantes segundo as atividades das etapas iniciais.

3.1 DESCRIÇÃO DAS FUNCIONALIDADES DE SOFTWARE RELEVANTES

Baseando-se na definição de Olsen, Samavati *et al.* (2009), as principais funcionalidades operativas de um software de modelagem baseada em sketch são: aquisição, reconhecimento, interpretação e reconstrução⁴². Adicionado a isto, este software deve oferecer funções de edição do modelo 3D reconstruído, suportando modos de edição que sejam relevantes para arquitetos, como por exemplo, a partir de representações 2D de elevações e cortes, oferecer funcionalidades de visualização geográfica e navegação de mapas.

Lembrando o contexto histórico e a adoção da tecnologia previamente estudados, faz-se importante considerar os critérios de design de interface estipulados pelos desenvolvedores dos sistemas operacionais destas tecnologias. Dentre os Princípios Android de Design (AOSP, 2014), é relevante o princípio **Conhece-me**: “um aplicativo deve aprender sobre as preferências do usuário com o tempo”. Em vez de seguir os mesmos protocolos uma e outra vez, o aplicativo deveria oferecer a seleção das escolhas prévias num acesso mais próximo.

Um princípio relacionado é: **Só me mostra o que preciso, quando preciso**⁴³: “subdivide tarefas e informação em parcelas pequenas e inteligíveis. Esconde opções quando não são essenciais no momento e as apresenta na medida em que a tarefa avança” (*op. cit.*). Isto pode ser implementado através de um menu contextual adaptável, que ofereça só as funções que façam sentido no elemento (geométrico, arquitetônico) para a qual o menu contextual estivesse sendo ativado.

Outro princípio relevante é: **Dê-me truques que funcionem em todas partes**: o aplicativo se faz fácil de aprender aproveitando padrões de interação e de “memória muscular”

⁴² Veja-se a explicação dos principais conceitos operativos dos softwares de modelagem baseada em sketch, na página 33, no ponto 1.2.5 Evolução da interação por caneta e *touch*.

⁴³ Tradução de “*Only show what I need when I need it*” (AOSP, 2014).

já utilizados em outros aplicativos. Por exemplo, o gesto de arrastar (*swipe*) é costumeiramente usado para navegação (*op. cit.*).

Estes princípios de design são recomendados por Google para desenvolvedores que publicam os seus aplicativos na loja virtual oficial Google Play. Já as Diretrizes iOS de Interface Humana (APPLE, 2014) abarcam todos os princípios anteriores e muitos outros, que são cumprimento obrigatório para a publicação de aplicativos na iTunes App Store (*ibid.*).

3.1.1 Interação arquiteto-computador

Para o correto avanço do usuário nas tarefas de elaboração do projeto, o software deve oferecer um feedback contínuo das tarefas bem sucedidas e dos possíveis erros e oportunidades de recuperação. Na ausência de uma linha de comandos, este feedback pode chegar ao usuário ainda na forma de mensagens escritas, mas estas mensagens não podem ser apresentadas de maneira a interromper o processo cognitivo do mesmo (AOSP, 2014). Para isto, pode ser desenvolvido um subsistema de chamada, que só apresente a mensagem quando ativada. Esta chamada deve ser codificada da maneira mais natural possível, segundo os recursos comunicativos representativos do tipo de trabalho, que neste caso se referem aos recursos visuais dos sketches comunicativos, como os definidos por Do (1999) e Forbus, Usher *et al.* (2008).

Estas funcionalidades operativas devem ser controladas a partir de um sistema de interação simples, que aproveite as convenções culturais construídas em torno das *affordances* dos instrumentos mais representativos da atividade a desempenhar (STILL e DARK, 2013), o que, no caso da elaboração de sketches, se refere às *affordances* do lápis e do papel⁴⁴, aumentados na substituição deles pelos dispositivos: caneta *stylus* e tela *multi-touch*.

Aproveitando as tecnologias de sensores de proximidade e de discriminação de instrumentos de entrada entre caneta e dedos, incluídas nas telas *multi-touch*, é possível estabelecer regras de inferência de modos operativos de edição e de navegação para cada tipo de instrumento, como em Negulescu, Ruiz e Lank (2010). Em outras palavras, estabelecer, com regras simples e intuitivas para o usuário, uma relação direta entre modos de interação e modos operativos: a caneta desenha, os dedos navegam e ativam mensagens, e a combinação simultânea de ambos modifica o modelo. Desta maneira, a interação pode ser articulada por gestos, segundo o vocabulário descrito na sequência.

⁴⁴ Ver *affordance* do lápis, na página 17, no ponto 1.1.1 Técnicas de desenho mais comuns nas etapas iniciais de projeto.

3.1.2 Definição de vocabulário de gestos

Baseado na fundamentação teórica e nos princípios e diretrizes dos sistemas operacionais previamente mencionados, define-se um vocabulário de gestos atribuído as funcionalidades de apoio às etapas iniciais de projeto.

No modo de **criação** de desenhos, ao ser definido pelo uso exclusivo da caneta *stylus*, limita-se o emprego de gestos para evitar conflitos cognitivos entre gestos similares, como demonstrado em Long (2001). Isto também garante a utilização mais natural possível, tendo como referência a *affordance* do lápis tradicional. O único gesto proposto é o de apagar: rabisco em ziguezague sobre o objeto a ser apagado, segundo definido por Fonseca e Jorge (2001), ilustrado na Figura 22 (página 57), e reaproveitado por Company, Contero *et al.* (2003), Naya, Contero *et al.* (2008) e Lopes, Mendes *et al.* (2011).

Para a **navegação**, é importante aproveitar o vocabulário de gestos *multi-touch* com o qual os usuários de computadores *tablet* estão familiarizados (VILLAMOR, WILLIS e WROBLEWSKI, 2010), com exceção das diferenças existentes entre plataformas e aplicativos em relação ao uso de múltiplos dedos nas tarefas mais simples. Por exemplo, a diferença entre o *Pan* definido pelo arrasto de um dedo em Google Earth e de dois dedos em Autodesk FormIt. A aproximação do FormIt permite uma navegação mais ágil em casos de zoom e *pan* simultâneos, mas diminui a percepção de segurança no controle do *Pan*. O vocabulário proposto para gestos de navegação é listado a seguir, na Tabela 6.

Tabela 6: Gestos de navegação propostos.

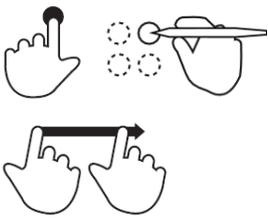
Funcionalidade	Gesto
Pan	 Arrastar
Zoom	 Pinçar/Estender
Orbitar	 Arrasto de dois dedos
Zoom ao objeto	 Toque duplo sobre objeto
Zoom à extensão	 Pinçar/Estender com mais de 2 dedos
Menu contextual	 Pressionar (tocar e segurar uns segundos)

Fonte: Proposta nossa, desenhos de Villamor, Willis e Wroblewski (2010).

Os gestos propostos na Tabela 6 definem as entradas de toque com os dedos para navegação, tanto em vistas ortogonais (planta, elevação), como em perspectiva, garantindo consistência no sistema de interação. A metade dos gestos de navegação (Zoom, Orbitar e Zoom à extensão) utiliza vários dedos. Os gestos de *Pan*, Zoom ao objeto e a chamada ao Menu Contextual são de apenas um dedo.

Para as operações de **edição** do modelo previamente criado, modificações nas ações de modelagem (como a extrusão manual ilustrada nos *storyboards* dos cenários de uso) e outras transformações definem-se gestos compostos. Alguns destes gestos são bimanuais multimodais, seguindo a divisão assimétrica de tarefas bimanuais de Guiard (1987), adaptadas para caneta e *multi-touch* por Hinckley, Yatani *et al.* (2010a) (2010b), como listado na Tabela 7, onde MD significa mão dominante e MND significa mão não dominante.

Tabela 7: Gestos propostos para transformações e edições avançadas.

Funcionalidade	Gesto	
Mover		Tocar objeto e arrastar.
Rotacionar		Segurar com MND e arrasto de dois dedos com MD sobre objeto.
Escalar		Segurar objeto com MND e arrastar com MD.
Copiar		Segurar objeto com MND e toque duplo com caneta no destino da cópia.
Extrusão manual		Segurar superfície com MND e desenhar linha de extrusão com caneta.
Cópia sequencial		Segurar objeto com MND, toque sequencial com caneta nos destinos da cópia, e arrasto para definir número de repetições.

Fonte: Proposta nossa baseada em Hinckley, Yatani et al. (2010a) (2010b), e Villamor, Willis e Wroblewski (2010).

Os gestos propostos na Tabela 7 implicam em sequência de toques, alguns só com dedos e outros com dedo e caneta. O mais simples é o de Mover, que pode ser feito com um só dedo. O uso exclusivo dos dedos limita-se as transformações básicas de um objeto já criado: Mover, Rotacionar e Escalar. O uso da caneta em gestos compostos destina-se a ações que implicam em mais criação: Copiar, Extrusão manual e Cópia sequencial (HINCKLEY, YATANI *et al.*, 2010a).

3.1.3 Processamento geométrico

O software deve oferecer a função para o usuário desenhar, um por um se assim o quiser, objetos das diferentes classes de elementos arquitetônicos sobre um terreno dado. Este desenho deve ser introduzido em forma de sketch, rabiscando na tela com uma caneta *stylus*, e reconstruído como geometria 3D da classe apropriada. A principal sequência (não exaustiva) de funcionalidades pode ser:

- Adquirir os traços desenhados pelo usuário como curvas ou linhas poligonais;

- Estruturar as linhas do desenho como um grafo;
- Reconhecer, no grafo do desenho, elementos geométricos (pontos, polígonos abertos, fechados e formas primitivas);
- Interpretar, nos elementos reconhecidos, objetos das classes de desenho no seu contexto (ex. em projeto urbano: eixos viários, ruas, calçadas, quadras, parcelas e edificações);
- Reconstruir os objetos do projeto num modelo em 3D;
- Simular fatores de desempenho;
- Oferecer funções de edição da geometria reconstruída/simulada: transformações euclidianas (mover, rotacionar), afins (escalar, cisalhar), e compostas (afunilar, torcer, duplicações em sequência).

No desenvolvimento destas tecnologias, especialmente em áreas de interação humano-computador e design de software, a elaboração de protótipos é uma prática comum (BEAUDOUIN-LAFON e MACKAY, 2002) (GREENBERG *et al.*, 2011). Estes podem ser de alta definição (ex.: fragmentos funcionais de software com aparência final) ou de baixa definição, que são especialmente úteis na etapa de planejamento de software, para a definição das funcionalidades a desenvolver. Entre os protótipos de baixa definição, destaca-se o *storyboard* não só como recurso de projeto, mas também como instrumento de estudo teórico-prático de hipóteses e cenários de utilização (*ibid.*). Com este intuito, apresentam-se na sequência, cenários de utilização ilustrados por *storyboards*.

3.2 ESTABELECIMENTO DE CENÁRIOS DE UTILIZAÇÃO

Dois cenários hipotéticos de utilização são elaborados explorando possibilidades de desenvolvimento de projetos de natureza diferente: um partido arquitetônico baseado num estudo volumétrico livre de contexto urbano, envolvido basicamente com atividades de modelagem geométrica (cenário 1); e o plano de ocupação de uma intervenção urbana, explorando aspectos geográficos, sua relação com o desenho arquitetônico e as diferentes interações entre eles (cenário 2).

O processo de elaboração dos desenhos a seguir, se fundamenta nos conceitos anteriormente estudados sobre o processo de pensamento do arquiteto e a tendência a representar os primeiros desenhos em duas dimensões (KALLIO, 2005) (ELSEN, DEMARET *et al.*, 2008). O projetista se apoia na sua inteligência visual e conhecimento implícito para interpretar nos sketches, as características tridimensionais do desenho em elaboração e,

posteriormente, reinterpretar a representação em 3D, levando a alterações e avanços de projeto (OXMAN, 1997).

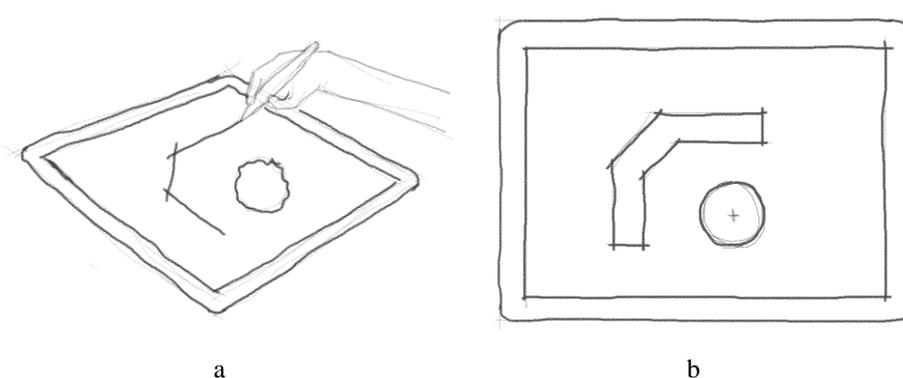
A seleção dos projetos a serem desenvolvidos pelo usuário, de determinada tipologia, porte (tamanho da edificação) e configuração geométrica, se fundamenta nos estudos do estado da arte, que demonstraram quais as características mais representativas, às quais um software desta natureza deveria responder com o suporte apropriado⁴⁵. Assim nos cenários, são exploradas geometrias predominantemente ortogonais, polígonos com segmentos de retas e curvas em arco, que formam volumes por gestos de extrusão (barrido linear na terceira dimensão).

3.2.1 Cenário 1

Neste cenário, o projetista possui informações básicas sobre necessidades espaciais e tem algumas ideias sobre possíveis partidos para experimentar.

Etapa 1: Lançamento de ideia de partido. O projetista abre o nosso aplicativo de projeto assistido por computador e a tela inicial já mostra uma folha nova para começar a desenhar e, nela, pensa nos componentes do programa de necessidades e a atribuição de formas primitivas a estes. Com a caneta, o usuário começa traçar rabiscos que são desenhados na folha em cores difusas de cinza, cujo valor (intensidade de cinza) depende da pressão que é exercida com a caneta sobre a tela (NAYA, CONTERO *et al.*, 2008) (Figura 27a).

Figura 27: *Storyboard* 1: traços iniciais de um sketch reconhecidos automaticamente como formas geométricas.



Esses rabiscos formam visualmente duas formas fechadas, que o projetista imagina como áreas de edificação que deixam um vazio entre elas e estabelecem certa relação de tensão. Assim como em Yu e Zhang (2007), o reconhecimento não é “on-line” (em tempo real), mas disparado pela inatividade. No momento em que o usuário afasta a caneta e as mãos da tela por

⁴⁵ Ver na página 44, no ponto 2.1 Configuração geométrica testada.

mais de quatro segundos, o software aciona o reconhecimento do sketch (processo que pode ser pausado se o usuário retoma o desenho do sketch). Logo depois, a folha mostra as geometrias reconhecidas numa cor marrom (cor por omissão, que pode ser alterado nas preferências do aplicativo), diferenciando-se dos traços do sketch, que se mantém nas cores cinza originais. No exemplo imaginado neste cenário, o usuário desenhou duas formas bidimensionais, que são reconhecidas como um círculo e um polígono fechado (Figura 27b).

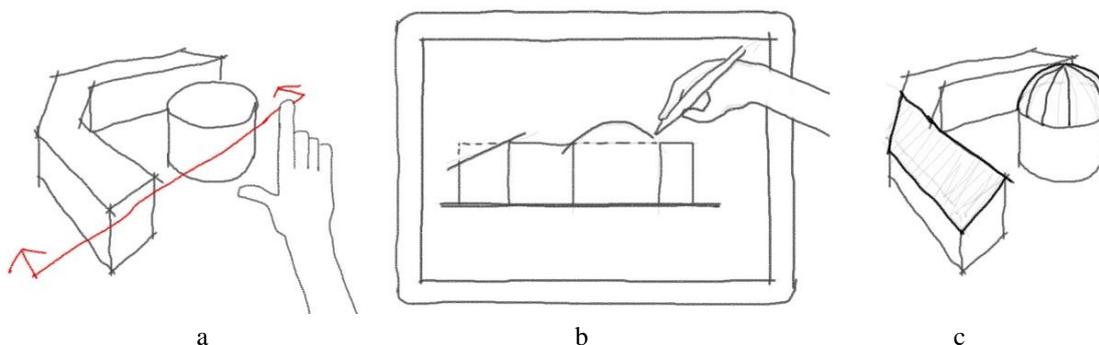
Seguidamente, o projetista estabelece relações de topologia e proporção na composição, entendendo-a como as futuras edificações. No momento em que se utiliza o gesto de Orbitar (rotacionar câmera 3D, centrando no desenho), o desenho muda da projeção 2D em planta à projeção em perspectiva cônica, na qual as formas reconhecidas aparecem como elementos planos, apoiados no chão (plano XY). O projetista define alturas dos volumes de edificação, para o qual seleciona os dois elementos com um toque simples neles e, com um gesto bimanual de extrusão (LOPES, MENDES *et al.*, 2011) (com a mão não dominante, um dedo fixo na tela; e com a dominante, o arrasto da caneta), os elementos planos geram sólidos estruídos na direção e no tamanho da linha de extrusão desenhada. O comando de extrusão ainda fica ativo, o que é visualmente explícito, já que as faces superiores dos sólidos ainda estão selecionadas e o eixo de extrusão ressaltado. Em caso do usuário querer modificar o tamanho da extrusão, ele deve empurrar as faces num ou noutro sentido da extrusão (aumentar ou diminuir de altura), similar à funcionalidade de Oh Stuerzlinger e Danahy (2006) e de Yu e Zhang (2007), mas usando o gesto de arrasto (Drag) (para sair do comando, basta tocar fora dos sólidos ou iniciar uma outra ação com a caneta).

O projetista faz uma revisão do modelo criado, navegando em torno deste com gestos *multi-touch*: Orbit, Zoom e Pan (KANG, KIM *et al.*, 2013). Após a breve revisão, o projetista gera alternativas através da modificação das formas e as relações entre elas. Empregando gestos multimodais, executam-se transformações como mover, rotacionar, escalar e esticar; em planta, para controlar as modificações das bases, e em perspectiva, para controlar as alturas (OH, STUERZLINGER e DANAHY, 2006) (YU e ZHANG, 2007). Tudo isto alternando com gestos *multi-touch* de navegação para verificar o efeito das transformações desde vários ângulos e níveis de aproximação.

Etapa 2: Definição de coberturas. Uma vez satisfeito com a configuração volumétrica, o projetista decide avançar na definição das edificações. Ele especifica a geometria das coberturas numa vista em corte, que deve definir. Para isto, desenha uma série de traços, que são reconhecidos pelo aplicativo como um conjunto de linhas poligonais localizadas no chão e atravessando os dois sólidos (Figura 28a). Pela sua topologia e localização, o conjunto é

interpretado como linha de corte e adquire a cor característica dos símbolos (vermelho por omissão), para diferenciá-la dos elementos em processo de modelagem.

Figura 28: *Storyboard 1*: Ativação da vista em corte, redesenho de coberturas e verificação das modificações em perspectiva.

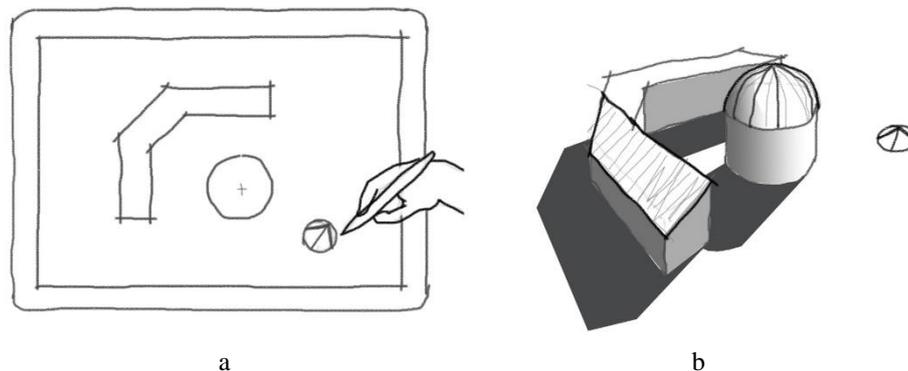


No momento em que o usuário faz um toque duplo na linha de corte, aciona o modo de edição do corte, para o qual o desenho muda à projeção bidimensional em elevação (neste caso, um plano paralelo ao plano XZ), mostrando o desenho cortado e as linhas visíveis segundo a direção que foi definida no símbolo de linha de corte. O usuário modifica a geometria mostrada, desenhando linhas (Figura 28b) e apagando as que estão sendo substituídas por elas, com o gesto de apagar (rabisco em ziguezague transversal à linha a apagar), como em Naya, Contero *et al.* (2008).

Fazendo o gesto de Orbital, o desenho volta à projeção em perspectiva, na qual é possível conferir que a edição das linhas de corte tenha modificado a geometria dos sólidos para acompanhar a nova condição geométrica recentemente definida (Figura 28c).

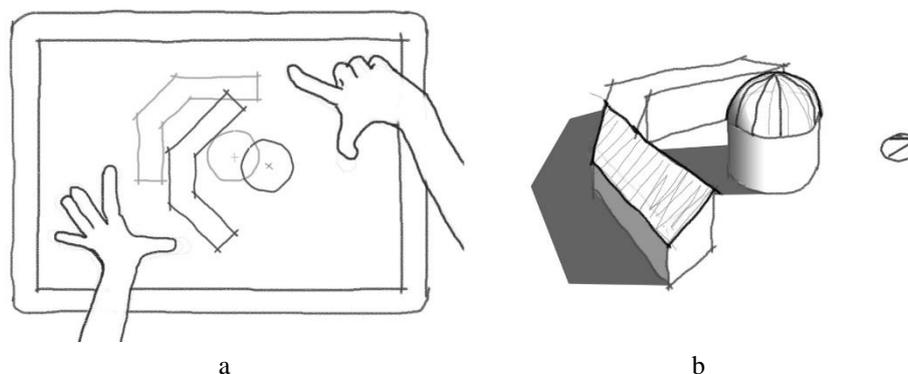
Voltando à vista em planta, o usuário desenha, num espaço vazio, a marcação do Norte (Figura 29a). No momento em que esse símbolo é reconhecido, sua topologia corresponde com a base de dados de símbolos, como em Juchmes, Leclercq e Azar (2005), e se interpreta como sistema coordenado de referência. Este dado permite simular o desempenho com respeito a fatores ambientais, como o sombreamento (TURKIENICZ, GONÇALVES e GRAZZIOTIN, 2008), acionando esta simulação com o toque de um botão que aparece na tela num menu contextual, quando o usuário usa o gesto de pressionar (segurar o toque por dois segundos). Este é um menu livre, como em Lopes, Mendes *et al.* (2011), que só oferece opções que fazem sentido para as transformações ou edições possíveis segundo as definições do modelo, como, neste caso, simulações de desempenho ambiental. Como o usuário não tem especificado data e hora para a simulação, o cálculo é feito para a data e hora atuais. Uma vez desenhada a simulação, o usuário navega pela composição em perspectiva para inspecioná-la (Figura 29b).

Figura 29: *Storyboard 1*: O desenho do rótulo Norte define orientação e permite simulações de desempenho ambiental.



Etapa 3: Ajustes preliminares de desempenho. Neste cenário, o projetista aproveita a visualização do sombreamento para abordar outro problema relacionado com a incidência de luz direta do sol em algumas fachadas (que por enquanto não são mais do que superfícies fechadas). Ao ver um problema de sombreamento do edifício cilíndrico sobre uma ala do edifício poligonal, decide mudar a orientação da composição, para o qual volta à vista em planta, seleciona os elementos da composição e os rotaciona (Figura 30a).

Figura 30: *Storyboard 1*: Rotacionando e orbitando o modelo recalcula-se a simulação de Sombreamento.

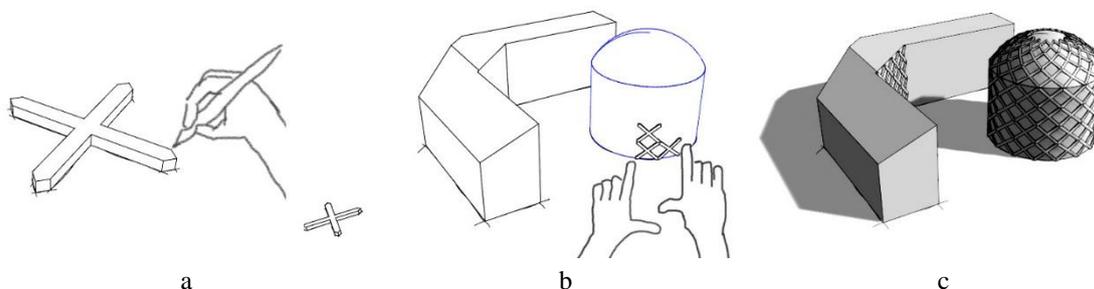


Uma vez acabada a transformação, o usuário afasta as mãos da tela por quatro segundos, fazendo com que o aplicativo recalcule a simulação de sombreamento (com uma rápida resposta, assim como em MARSY e LIPSON, 2005) (Figura 30b). Uma vez restaurada a simulação, o projetista verifica que a sombra agora se projeta sobre a ala central, na qual, pelos requerimentos de conforto dos espaços que estarão ali, é menos problemática.

Etapa 4: Composição de fachada. O projetista aborda a estruturação do invólucro a partir de uma modulação. Num espaço vazio, desenha um polígono em forma de X, e lhe aplica uma espessura com uma pequena extrusão (Figura 31a). Por meio de gestos bimanuais, como em Hinkley, Yatani *et al.* (2010a) (2010b), copia a forma num arranjo de matriz sobre as superfícies

de um dos edifícios (Figura 31b). Após a transformação, segundos de inatividade dispararam o recálculo da simulação de sombreamento (Figura 31c).

Figura 31: *Storyboard 1*: Composição de fachadas através do desenho e transformação avançada de um módulo, utilizando gestos bimanuais.



As experimentações formais podem ser prolongadas consistindo em várias iterações de edição geométrica e verificação de sombreamento, até o projetista estar satisfeito com o partido arquitetônico desenvolvido. Então, pode exportar o modelo para outros aplicativos e continuar o desenvolvimento do projeto nas etapas seguintes, como o estudo preliminar e o anteprojeto, assim como em Company, Contero *et al.* (2003) e Naya, Contero *et al.* (2008).

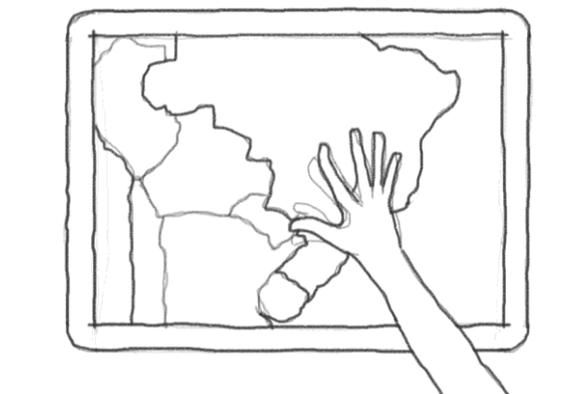
3.2.2 Cenário 2

Etapa 1: Definição de contexto geográfico e urbano. O projetista abre o nosso aplicativo e, na tela inicial, toca e segura no espaço vazio, chamando o menu contextual. Dentro deste menu, seleciona a opção de definir uma localização geográfica. Em seguida, aparece na tela o mapa terráqueo centrado no lugar que marca a atual localização geográfica do usuário (segundo a informação de GPS ativa, ou a informação de localização da conexão de internet. Caso o computador esteja desconectado da internet, se usa a última localização registrada). O usuário navega no mapa terráqueo com o toque dos dedos (Figura 32) para deslocar a posição (*Pan*) e se aproximar (*Zoom*) à localização do terreno sobre o qual vai desenvolver um projeto de intervenção urbana. No momento em que navega pelo mapa, a imagem vai se atualizando em tempo real, mostrando, quando a escala o permite, informação de acidentes geográficos, estradas, limites geopolíticos, centros urbanos e, no canto inferior direito, a escala de aproximação (escala gráfica basicamente. Quando o zoom atinge alguma escala padrão, como 1:25.000 ou 1:5.000, é também acompanhada desta escala numérica). Uma vez que o usuário tenha encontrado a localização que buscava, define o ponto de localização, marcando-o com uma função específica, acessada pelo menu contextual.

Quando o usuário coloca sobre a tela a ponta da caneta, o aplicativo ativa o modo de edição: muda a visualização do terreno da vista 3D à vista 2D em planta, sobrepondo uma

camada branca semitransparente acima da imagem fotográfica, sobre a qual o usuário pode desenhar sketches, assim como em Pranovich (2004) (a vista em planta, quando acionada após a visualização de um terreno em 3D sobrepõe esta camada branca semitransparente para garantir suficiente contraste de cinzas sobre branco no desenho de sketches. A cor e o nível de transparência da camada de fundo das vistas de sketch são customizáveis).

Figura 32: *Storyboard 2*: Navegação *multi-touch* pelo mapa terráqueo para definição de localização do projeto.

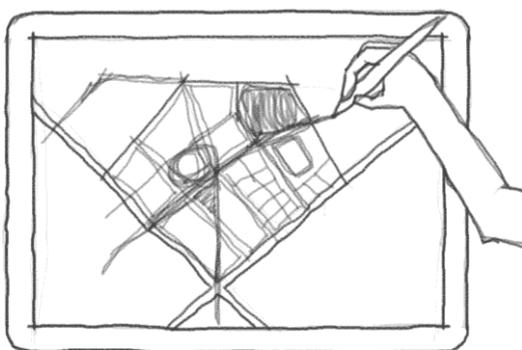


O projetista, usuário do aplicativo, desenha com a caneta um polígono acima de um lote que identifica através da fotografia aérea mapeada acima da mesh do terreno 3D. Os traços são desenhados acima da imagem do terreno, adquiridos como linhas, e reconhecidos como um polígono na camada 2D posicionada acima do terreno 3D. Sobre o polígono, o usuário toca e segura o toque por dois segundos, até que emerge o menu contextual, dentro do qual seleciona a opção de definição de perímetro de terreno (este menu contextual só oferece opções que fazem sentido para as transformações ou edições possíveis no elemento selecionado (LOPES, MENDES *et al.*, 2011). Por ser um polígono predominantemente horizontal, de dimensões maiores do mínimo admissível para um lote de terreno, o menu oferece a opção de atribuir o metadado que define “Perímetro limítrofe” como classe de elemento urbano. Para isto é necessário que o software calcule as dimensões do polígono e avalie a sua horizontalidade e tamanho mínimo, assim como em Juchmes, Leclercq e Azar (2005). Isto é possível neste caso, já que o usuário começou o seu trabalho a partir da navegação do mapa terráqueo, o que definiu a localização, orientação e escala. Ao fazer isso, o aplicativo pergunta ao usuário se ele quer incluir as construções existentes dentro do polígono ou desconsidera-las. O usuário opta por desconsiderar as construções, com o qual o software reconstrói a *mesh* do lote, alisando o terreno. O projetista apenas percebe a modificação da topografia, já que faz estas ações sem sair da vista 2D em planta, mas percebe a modificação do conteúdo, devido a que a novo lote

tem uma coloração constante, diferente daquela que utilizava a fotografia aérea e que ainda define a aparência do terreno de entorno.

Etapa 2: Definição de elementos urbanos. O projetista retoma a atividade de desenho do sketch, rabiscando vários traços de dimensões e pesos diversos. O software identifica, pela morfologia e densidade dos traços, que se trata de um sketch de pensamento (COMPANY, CONTERO *et al.*, 2006), no qual o projetista está usando o feedback do sketch para abordar problemas mal definidos. Por isto, o software mantém visível a representação do sketch, em vez da representação da geometria sendo interpretada (Figura 33).

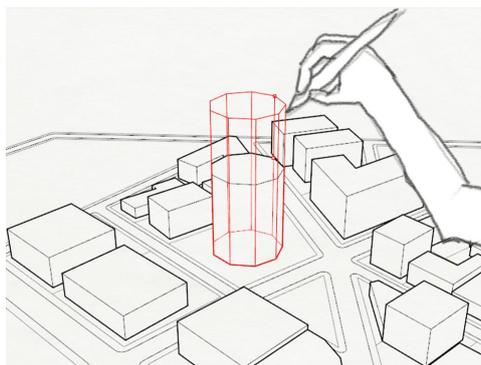
Figura 33: *Storyboard 2*: Desenho de um sketch de pensamento sendo adquirido pelo software sem interrupções.



Pela escala do sketch, os traços são interpretados como elementos de desenho urbano: eixos de vias, perímetro limítrofe, vias, quadras, calçadas, parcelas e silhuetas de edificações (*footprints*). Cada vez que o usuário para de desenhar, se dispara o processo de reconhecimento, interpretação e reconstrução: (i) reconhecem-se as formas, definindo os polígonos abertos e fechados que são visualmente emergentes; (ii) interpreta-se a qual tipo de elemento de desenho urbano previamente mencionado pertencem; e (iii) reconstrói-se a geometria, projetando as formas sobre a malha do terreno 3D e estudando-as às alturas padrão para cada tipo de elemento, assim como em Elsen, Demaret *et al.* (2012). Logo, calculam-se e exibem-se as sombras em planta.

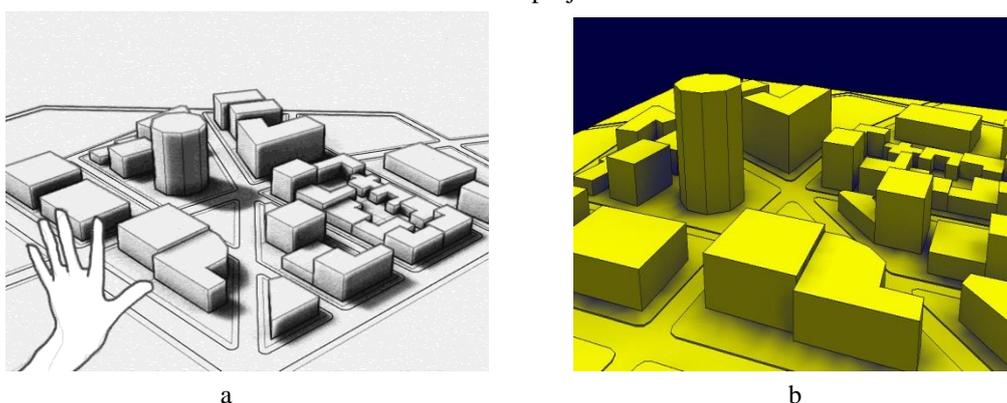
Quando o usuário afasta a caneta *stylus* da tela, o aplicativo volta ao modo de navegação, no qual se pode mover (*Pan*) e rotacionar a vista (*Orbitar*). Ao orbitar a partir da vista em planta, a projeção muda automaticamente da plana à perspectiva. Uma vez em perspectiva, o usuário pode voltar ao modo de edição apontando com a caneta. Assim, neste cenário imaginado, o usuário edita a altura dos edifícios reconstruídos para satisfazer uma condição específica que ele quer simular: o impacto de conforto que um edifício alto tem sobre os seus vizinhos mais baixos, puxando a face de cobertura no sentido da extrusão (Figura 34), como em Oh, Stuerzlinger e Dahany (2006), e em Yu e Zhang (2007).

Figura 34: *Storyboard 2*: Edição manual de uma extrusão, para modificar a altura de um edifício reconstruído.



Etapa 3: Simulação preliminar de desempenho ambiental. Para avaliar o impacto no conforto, o projetista se vale das funcionalidades de simulação de desempenho de alguns fatores ambientais. O primeiro a ser simulado é o de sombreamento (TURKIENICZ, GONÇALVES e GRAZZIOTIN, 2008), que é mostrado na tela após o usuário visualizar o modelo em perspectiva e afastar as mãos da tela por mais de quatro segundos (Figura 35a). Para simular outros fatores ambientais, o usuário deve chamar a execução de outra simulação através do menu contextual que aparece na tela com o gesto de tocar e segurar. Neste caso, o usuário aciona a simulação de iluminância para o presente ano (valor por omissão), espera uns segundos pelo cálculo e construção da simulação e, uma vez mostrada na perspectiva, continua a navegação pelo modelo, que exibe uma nova codificação de cores segundo o resultado da análise da iluminância (Figura 35b).

Figura 35: *Storyboard 2*: Simulação de sombreamento (a) e de iluminância (b) sobre os edifícios e o espaço urbano sendo projetado.



Este processo de modificação geométrica e a atualização das simulações de desempenho se repetem várias vezes até que o projetista considere que tenha atingido uma solução aceitável, após a qual pode exportar o modelo a outros aplicativos e usá-lo para avançar no processo de projeto nas etapas avançadas. Por exemplo, exportando a um aplicativo SIG no qual possa gerar

mapas e documentação sobre o projeto de intervenção urbana; e exportando a um aplicativo BIM, os volumes de ocupação das edificações (*massing*).

3.2.3 Síntese de funcionalidades sugeridas nos cenários de utilização

A Tabela 8, reúne as interações e ações de processamento sugeridas nos cenários de uso acima descritos. Nesta tabela, listam-se em função dos processos cognitivos do projetista.

Ao mensurar a informação representada na Tabela 8, é possível observar certa desproporção entre a quantidade de processos cognitivos iniciados pelo projetista, a quantidade de gestos e ações de interação e a quantidade de funcionalidades de processamento disparadas no software em função desta atividade. Nas primeiras etapas, acontecem muitos processos cognitivos, que levam o projetista a executar poucas ações de interação, e iniciam uma menor quantidade de ações de processamento por parte do software⁴⁶.

⁴⁶ Nos referimos a ações de alto nível de abstração. Estas ações, de fato, dependem de milhares de processos de computação de baixo nível, como o cômputo numérico, irrelevantes na etapa de planejamento de software.

Tabela 8: Interações e processamento segundo processos cognitivos, sugeridos nos cenários de uso.

Processos cognitivos na projeção (O projetista pensa em:)	Gesto/Interação (O projetista atua:)	Funcionalidade/Processamento (O software responde com:)
Atribuição de formas primitivas a componentes do programa de necessidades.	Desenho com caneta; apagar; redesenhar.	Aquisição de traços.
Estabelecimento de relações iniciais de topologia e proporções em planta.	Segundos de inatividade na tela;	Fechamento de polígonos; reconhecimento de formas emergentes.
Estabelecimento de relações de topologia e proporções em volumetria; o projetista define alturas dos volumes gerados; imagina, nestes volumes, edifícios; e reavalia as relações de topologia e proporção.	Gestos <i>multi-touch</i> de navegação; gesto de Extrusão.	Navegação: Orbital, <i>Pan</i> , <i>Zoom</i> ; construção de sólidos por Extrusão.
Geração de alternativas; Modificação de relações entre formas em 2D e entre volumes de edificação; avaliação comparativa.	Gestos <i>multi-touch</i> de navegação; gestos multimodais de transformações.	Transições entre vistas em planta e em perspectiva; transformações: mover, rotacionar, escalar, esticar.
Arbitrar posicionamento estratégico de corte transversal; modificação de coberturas a partir da vista em corte; verificação da modificação.	Desenho de símbolo arquitetônico; toque duplo; redesenho de sketches; gestos <i>multi-touch</i> de navegação.	Reconhecimento de símbolo arquitetônico; ativação de vista em corte; aquisição de traços, reconhecimento, interpretação como elementos de cobertura, reconstrução do modelo.
Definição de orientação com respeito ao Norte.	Desenho de símbolo arquitetônico; segundos de inatividade na tela.	Reconhecimento de símbolo arquitetônico; reajuste da orientação do plano de trabalho.
Verificar problemas de sombreamento entre edificações.	Menu contextual; botão de simulação de Sombreamento; gestos <i>multi-touch</i> de navegação.	Amostragem da mesh; cálculo de Oclusão e Sombreamento; atribuição de cores de vértices, segundo valores de oclusão; navegação: Orbital, <i>Pan</i> , <i>Zoom</i> .
Modificações globais para solucionar problemas de sombreamento entre edificações.	Gestos <i>multi-touch</i> de navegação; Gesto multimodal de transformação; segundos de inatividade.	Transições entre vistas em planta e em perspectiva; transformação: rotacionar; recálculo de Oclusão e Sombreamento; redefinição de cores de vértices, segundo novos valores de oclusão; navegação: Orbital, <i>Pan</i> , <i>Zoom</i> .
Composição de fachadas em função à incidência solar; definição de um módulo de janela a partir de uma vista em elevação.	Desenho de símbolo arquitetônico; toque duplo; desenho de sketches.	Reconhecimento de símbolo arquitetônico; ativação de vista em elevação; aquisição de traços, reconhecimento, interpretação como elementos de fachada, reconstrução do modelo.
Repetição do módulo de janela ao longo das superfícies de fachada.	Gestos <i>multi-touch</i> de navegação e seleção; gestos multimodais de transformações avançadas.	Reparametrização de superfícies; duplicação em matriz sobre superfícies.

Fonte: o autor.

Na medida em que se avança na projeção, a quantidade de gestos e, especialmente, de processamento computacional aumenta consideravelmente em comparação com a quantidade de processos cognitivos do projetista. Isto não significa que o projetista diminui no aprofundamento da projeção, mas que as decisões de projeto são cada vez mais carregadas de conhecimentos implícitos, requerendo menores descrições para maiores definições formais e funcionais. Por exemplo, na sub-etapa de atribuição de formas primitivas a componentes do programa de necessidades, o projetista pensa em círculos, retângulos e outros polígonos, nas relações compositivas entre eles, e na direção e sentido de desenho de tais formas. Porém, o esforço do computador se centra em reconhecer formas primitivas e mostrá-las na tela. Por outro lado, nas modificações globais para solucionar problemas de sombreamento entre edificações, o projetista pensa nas mudanças de orientação geral do conjunto e no gesto para controlar a modificação. Neste caso, o computador deve reconhecer um gesto bimanual, fazer a transição de vistas em perspectiva e em planta, executar a transformação geométrica, recalcular o vetor do Sol, recalcular a oclusão e sombreamento, redefinir cores de vértices e redesenhar a visualização do modelo. Estas desproporções entre quantidade de processos relevantes em diferentes etapas de projeção devem ser levadas em consideração no desenvolvimento das funcionalidades de software, para que o projetista não perca a continuidade na atividade projetual.

As funcionalidades propostas estabelecem uma diferenciação entre os conceitos de reconhecimento, interpretação e reconstrução, inclusive entre tipos de reconstrução. Não se segue o princípio de reconstrução gestual em detrimento de uma reconstrução automática. Em vez de serem antagônicos, como sugere Company, Contero *et al.* (2006), propõe-se que sejam complementares e interdependentes entre eles. Quando há suficiente informação contextual (como nos edifícios no cenário 2), a interpretação pode ser automática. Quando falta esta informação, o software deve facilitar as possibilidades de interação do usuário para especificá-la manualmente (como no lote do terreno no cenário 2), mas também deve permitir o trabalho com interpretações incompletas (como no cenário 1).

Na Tabela 9, apresentada em seguida, se listam, de maneira sintética, as funcionalidades que deve oferecer um software na utilização imaginada nos cenários acima descritos. Independentemente de serem projetos de diferente natureza, aproveitam funcionalidades em comum. Tenta-se alinhar as ações de acesso, uso e processamento segundo a funcionalidade principal que as engloba, de maneira a serem comparáveis lado a lado.

O alinhamento lado a lado na Tabela 9 permite observar que o usuário omitiu algum passo em um ou outro cenário, sem que isso o impedisse de realizar atividades projetuais

satisfatoriamente. No cenário 1, o usuário começa diretamente com a elaboração do sketch, omitindo etapas de definição de localização geográfica descritos no cenário 2. Sem esta informação, o modelo atribui valores por omissão para a localização e escala, de maneira a poder definir coordenadas de arrasto do ponteiro para a aquisição dos traços do sketch.

No cenário 2, a primeira ativação do modo de edição interpõe uma camada de “papel croquis” (branco translúcido), não só para facilitar a visualização do sketch a ser elaborado, mas também para lembrar ao usuário que, mesmo gerando geometria em 3D, as atividades de desenho arquitetônico acontecem principalmente em 2D, assim como acontecem no desenho com lápis e papel. Isto para garantir uma familiaridade de uso e minimizar a necessidade de aprendizagem de novos protocolos em projetistas já treinados em desenho arquitetônico.

As sequências de reconhecimento, interpretação e reconstrução se ativam automaticamente após segundos de inatividade, uma vez que exista algo para reconhecer, mas estes processos podem ser interrompidos ou forçados a acontecer de uma maneira específica, como descrito na definição do perímetro limítrofe no cenário 2. Em outras palavras, há uma interpretação automática, mas o usuário pode dizer ao software como interpretar elementos do sketch. Esta combinação de processamento procedural e interativo também está presente no momento da reconstrução, descrita nos cenários através da extrusão de polígonos. Esta extrusão é automática num primeiro momento, atribuindo valores padrão segundo o elemento interpretado, mas na vista em perspectiva (ativada com o gesto de Orbitar) é possível utilizar a funcionalidade de extrusão manual.

Tabela 9: Funcionalidades necessárias para suportar as atividades de cada cenário de uso.

Funcionalidades	Cenário 1	Cenário 2
Papel em branco	Tela inicial.	Tela inicial.
Menu contextual	—	Tocar e segurar.
Buscar Localização	—	Botão Mapa no Menu contextual; A partir de localização predeterminada (GPS/IP); Navegar (<i>Pan</i> e <i>Zoom</i>).
Escala gráfica/numérica	—	Símbolo/Texto atualizado segundo o <i>Zoom</i> .
Definir Localização	—	Menu contextual; Botão Definir Localização.
Ativar o Modo de Edição 2D	Aproximação da caneta (<i>hover</i>).	Aproximação da caneta (<i>hover</i>); Ativação camada semitransparente.
Aquisição de Traços	Registro das coordenadas de arrasto do cursor e pressão de toque.	Registro das coordenadas de arrasto do cursor e pressão de toque.
Feedback Aquisição (Representação do Sketch)	Desenho em tempo real dos traços em escala de cinza, segundo pressão exercida.	Desenho em tempo real dos traços em escala de cinza, segundo pressão exercida.
Reconhecimento de formas	Após segundos de inatividade: Cálculo de interseções; Estruturação de árvore; Definição de linhas poligonais abertas e fechadas; Detecção de círculos e arcos.	Após segundos de inatividade: Cálculo de interseções; Estruturação de árvore; Definição de linhas poligonais abertas e fechadas; Detecção de círculos e arcos.
Interpretação	Interpretação automática: Definição de formas reconhecidas como superfícies planas no chão (plano XY em $Z=0$).	Interpretação manual: Toque de seleção; Menu contextual; Definição de Perímetro limítrofe. Interpretação automática: Definição de elementos urbanos e silhuetas de edificações.
Reconstrução	Massa edificada: Extrusão de superfícies à altura padrão.	Terreno inscrito: Projeção do polígono no terreno 3D; Solicitar Opção de reconstrução do terreno; Alisar mesh de terreno segundo altura de vértices do perímetro. Massa edificada: Extrusão de quadras e silhuetas de edificações nas alturas padrão.

Funcionalidades	Cenário 1	Cenário 2
Navegação 3D	Gesto Orbital 3D; Ativação da visualização do fundo 3D; Mudança de projeção paralela à cônica; Rotação de câmera.	Gesto Orbital 3D; Ativação da visualização do fundo 3D; Mudança de projeção paralela à cônica; Rotação de câmera.
Extrusão manual	Seleção de superfícies (Toque); Gesto bimanual de Extrusão; Ajuste do comprimento de extrusão por Arrasto.	Seleção de superfícies (Toque); Gesto bimanual de Extrusão; Ajuste do comprimento de extrusão por Arrasto.
Criar de Linha de Corte	Aquisição de Traços; Reconhecimento de Linha de Corte por similaridade com simbologia padrão; Atribuição de cor de símbolo à Linha de Corte.	—
Ativar Vista em Corte	Toque duplo na Linha de Corte; Rotação da câmera; Mudança de projeção cônica à paralela; Ativação de nova camada semitransparente.	—
Modificar modelo por Edição do corte	Aquisição de Traços; Reconhecimento de formas; Interpretação de superfícies adicionadas às existentes; Reconstrução da geometria.	—
Definir Orientação	Aquisição de Traços; Reconhecimento de formas; Interpretação de símbolo Norte; Redefinição do Sistema Coordenado.	—
Simular Sombreamento	Definição do Vetor do Sol para data e hora atuais; Geração de mesh amostrada; Cálculo de Oclusão; Atribuição de cores à mesh segundo valores de Oclusão.	Definição do Vetor do Sol para data e hora atuais; Geração de mesh amostrada; Cálculo de Oclusão; Atribuição de cores à mesh segundo valores de Oclusão.
Simular Iluminância	—	Definição de valores por omissão para o ano, cobertura de céu e materiais; Análise de Iluminância; Atribuição de cores à mesh segundo valores de Iluminância.
Modificar modelo por Transformação em planta	Menu contextual: Ativar visualização em Planta; Selecionar formas; Gesto rotacionar.	—
Transformações Avançadas	Toque de seleção; Gestos multimodais.	—
Exportar	Exportação do modelo 3D a formato BIM ou CAD.	Exportação de polígonos, propriedades e metadados a formato SIG; Exportação do modelo 3D a formato BIM.

Fonte: o autor.

Logo, descrevem-se os passos elaborados pelo projetista, no cenário 1, para modificar características geométricas do modelo não definíveis em planta, como a criação de coberturas diferenciadas, através do desenho em corte. Devido ao nível de detalhe requerido pela natureza do projeto, isto não é necessário no cenário 2.

Posteriormente, descrevem-se interações com simulações de desempenho. Devido à falta de informação sobre orientação no cenário 1, o usuário se vê na necessidade de especificá-la através do desenho do símbolo do Norte que, após a sua interpretação, recalcula o sistema de coordenadas do modelo. Uma vez tendo a informação apropriada, ativa-se a rotina de análise e simulação, que é a mesma para os dois cenários.

A rapidez de visualização das simulações de desempenho permite sua utilização como componente de validação das propostas em desenvolvimento, trazendo objetividade nas tomadas de decisão dentro do ciclo desenhar, inspecionar, revisar, sem perder a flexibilidade de desenho característica das etapas iniciais de projeto.

Por último, menciona-se a possibilidade de exportação para outros formatos, podendo dar continuidade ao desenvolvimento dos projetos. As diferentes possibilidades de exportação dependem da diversidade de informações definidas nesta primeira etapa. Os modelos 3D podem ser exportados em formatos CAD e, junto com os metadados e propriedades interpretadas, exportados em formatos BIM. A definição de informações geográficas do cenário 2 permite, além do BIM, a geração de arquivos SIG.

O processamento requerido para a aquisição e reconhecimento de sketches é distinto, dependendo do tipo de sketch que o usuário elabora: sketches prescritivos podem ser adquiridos segundo a continuidade temporal da inserção de traços de sketch e reconhecidos segundo as interseções entre estes traços (elemento representativo da expressão arquitetônica). Sketches de pensamento tem maior dificuldade de reconhecimento; devem ser adquiridos na sua totalidade para logo poder reconhecer formas emergentes. Pela sua natureza difusa, são sketches de traços sobrepostos e incompletos, a partir dos quais é necessário diferenciar uns conjuntos de traços dos outros e construir uma curva representativa do conjunto – a curva emergente da sobreposição –, para logo computar a formação de figuras emergentes com algoritmos como o de Gero e Yan (1993).

Sejam originários de sketches de pensamento ou de sketches prescritivos, uma vez reconhecidos os elementos geométricos – interseções, vértices, segmentos de retas, linhas poligonais, arcos e outras curvas, abertas e fechadas –, passa-se às funções de **interpretação**. Nas funções de interpretação discriminam-se elementos de edificação (ex.: polígonos de silhuetas de edifícios, linhas de projeção, linhas de divisa) e elementos indicativos (ex.: setas,

linhas de chamadas, texto). Conforme forem interpretados, os traços do sketch “alimentam” (são utilizados como argumento para) outras funções que definem o contexto e a semântica do modelo arquitetônico. Assim, elementos de edificação são reconstruídos em 3D, segundo as propriedades definidas pelo seu contexto geométrico (ex.: plano de construção, relação de escala entre elementos) (JUCHMES, LECLERCQ e AZAR, 2004) e pelas informações definidas pelos elementos indicativos (YU e ZHANG, 2007).

A principal função de **reconstrução** é a extrusão linear no sentido perpendicular ao plano de construção do sketch. Isto pela típica maneira de elaboração de sketches arquitetônicos (YU e ZHANG, 2007), na qual o projetista começa desenhando em 2D, geralmente em planta, para logo inspecionar e revisar o modelo em 3D (*op. cit.*).

É nesta revisão do modelo em 3D onde se faz relevante a funcionalidade de **simulação de desempenho**. Após o usuário utilizar gestos *multi-touch* para navegar e inspecionar o modelo 3D, é importante executar automaticamente a simulação de desempenho, disparada por um elemento indicativo, como ilustrado no cenário 1, ou pela pausa após gestos de navegação, como ilustrado no cenário 2.

A chamada da simulação executa outra reconstrução geométrica no fundo: a discretização do modelo em malhas poligonais subdivididas a uma distância de amostragem tal, que produz uma densidade homogênea de vértices sobre os quais é calculada a análise. O primeiro fator ambiental a simular é o de sombreamento, por ser o menos complexo e por servir de base para outros como, por exemplo, a radiação solar.

O processamento geométrico, aqui descrito de maneira conceitual, é deduzido das ações descritas nos cenários de utilização, apoiadas pelo embasamento prévio. Note-se a ênfase feita às ações projetuais do arquiteto, como elementos definidores, não só dos modos de interação, como também do tempo de execução admissível para não quebrar a continuidade nas ações projetuais. Esta ênfase deve prevalecer nas posteriores etapas de desenvolvimento de software, para que este possa dar uma contribuição relevante à área de projetos arquitetônicos.

4 CONCLUSÕES

A contextualização teórica dos problemas de projeto e dos processos cognitivos envolvidos na projeção arquitetônica apoiou a definição das ações projetuais mais relevantes para receber suporte de aplicações tecnológicas de software e hardware. Partindo do uso mais representativo das tecnologias computacionais entre arquitetos, apresentou-se uma descrição dos problemas na utilização de softwares CAD e hardwares instrumentados por teclado e mouse. Foi ressaltada a difícil relação entre projetistas e sistemas computacionais durante a elaboração das ideias iniciais de projeto. A falta de correspondência entre as necessidades de apoio nas etapas iniciais de projeto e os softwares disponíveis para tal fim foi evidenciada no estudo do estado da arte relacionado ao desenvolvimento de softwares experimentais para assistência nestas etapas.

A contextualização histórica das tecnologias digitais situou o estado atual da arte e fundamentou o argumento sobre a maturidade e a disponibilidade das tecnologias de entrada direta: canetas *stylus* e telas *multi-touch*. Assim, demonstrou-se que estes dispositivos são os mais adequados para instrumentar a interação entre o projetista e o computador nas etapas iniciais de projeto (COMPANY, CONTERO *et al.*, 2004) (JUCHMES, LECLERQ e AZAR, 2004) (MASRY e LIPSON, 2005) (NAYA, CONTERO *et al.*, 2008) (HINCKLEY, YATANI *et al.*, 2010a) (2010b).

O presente estudo, além de atualizar o conhecimento descrito por Olsen, Samavati *et al.* (2009), é o primeiro a concentrar ênfase nas aplicações tecnológicas de assistência nas etapas iniciais de projetos de arquitetura. Isto foi feito reconhecendo a relevância das aplicações desenvolvidas especificamente para este campo do conhecimento, sem ignorar os aportes das experiências pensadas para outros ramos do design e das engenharias, com conhecimentos técnicos também aproveitáveis para a instrumentação e metodologias de projeto de arquitetura.

Com respeito aos métodos computacionais, destacou-se a relevância da modelagem baseada em sketch e das simulações de desempenho. Em comparação com métodos CAD, são identificados como métodos de modelagem e avaliação mais adequados para as etapas iniciais de projeto (MASRY e LIPSON, 2005) ao permitirem o suporte computacional aos processos cognitivos do projetista. Adicionalmente, permitem suporte à externalização das representações mais apropriadas para abordar os problemas mal definidos destas etapas de projeto (KALLIO, 2005).

Com respeito ao auxílio na transição de representações e ideias de projeto, das etapas iniciais às etapas avançadas de projeto, se destacou a importância da interoperabilidade através

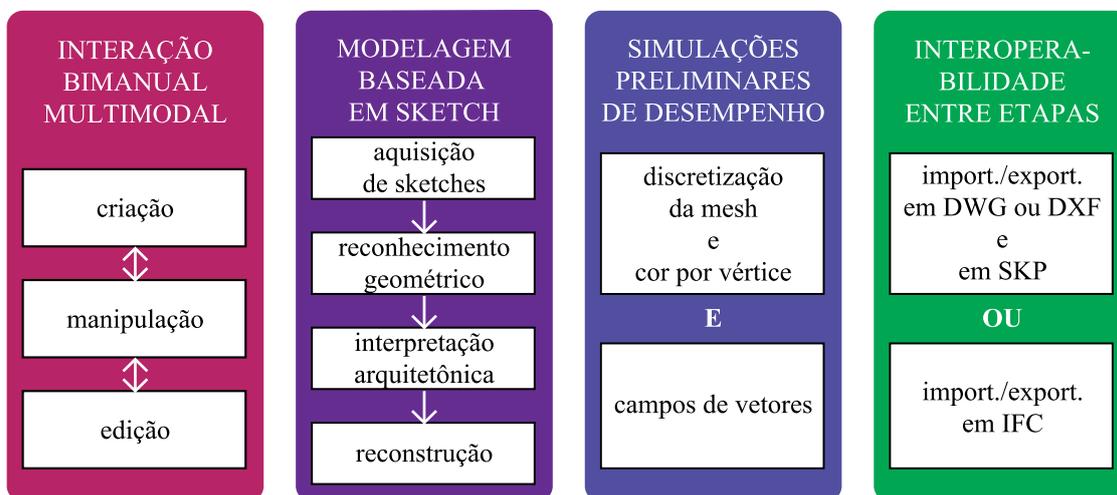
do suporte de importação e exportação de arquivos nos formatos mais relevantes para as etapas avançadas de projeto.

Os cenários de uso permitiram a exploração dos argumentos acima sintetizados e a especificação de funcionalidades de interação e de processamento. Ao verificar a possibilidade de continuidade entre funcionalidades combinadas, constatou-se que, ao se instrumentalizar um software que integre as funcionalidades requeridas, se proporciona notável agilidade as atividades das etapas iniciais de projeto arquitetônico.

Para concluir, destacam-se quatro principais conjuntos de princípios de funcionalidade, resumidos na Figura 36: **interação bimanual multimodal**, **modelagem baseada em sketch**, **simulações preliminares de desempenho** e **transição entre etapas de projeto**. Estes princípios são brevemente descritos a seguir, como critérios para a definição de requisitos de design e desenvolvimento de softwares para auxiliar arquitetos trabalharem de forma ágil e contínua durante as etapas iniciais de projeto.

4.1 PRINCÍPIOS DE FUNCIONALIDADE PARA APOIAR AS ETAPAS INICIAIS DE PROJETO ARQUITETÔNICO

Figura 36: Princípios de funcionalidade necessários em softwares de apoio nas etapas iniciais de projeto arquitetônico.



Fonte: O autor.

4.1.1 Interação bimanual multimodal

Implementar modos de interação inspirados no trabalho manual das mídias tradicionais, com dispositivos de entrada direta: caneta *stylus* e tela *multi-touch*. Combinada, esta instrumentação deve oferecer três modos de interação:

(i) **Criação:** desenho e anotação com a caneta;

(ii) **Manipulação:** navegação através do modelo do projeto utilizando gestos *multi-touch* (para funções de zoom, *pan*, rotacionar vista, selecionar, etc.);

(iii) **Edição:** modificar o modelo criado, por meio da combinação simultânea de caneta e dedos que define gestos bimanuais para funções de edição (copiar, cortar, colar, etc.) e transformações geométricas (mover, rotacionar, escalar, etc.).

Na necessidade de escolha de implementação de funcionalidades, cujos gestos sejam conflitivos, provindo uns de softwares preexistentes e outros de ações manuais, típicas da projeção arquitetônica, considerar os últimos.

4.1.2 Modelagem baseada em sketch

Basear a criação e edição de desenhos de projeto em traços introduzidos à mão livre no espaço de trabalho bidimensional, formando sketches que geram modelos em 3D. A criação deve ser baseada em quatro módulos de funcionalidades:

(i) **Aquisição de sketches:** obter, da entrada do usuário que utiliza uma caneta *stylus*, dados vetoriais do traço na tela como sequências de linhas. Podem-se registrar as variações de pressão imprimida com a caneta sobre a tela, para futuro uso;

(ii) **Reconhecimento geométrico:** o software deve analisar os sketches adquiridos para registrar formas primitivas e outros dados básicos. Essencialmente, precisa distinguir entre traços que representam geometria da edificação sendo projetada, e traços que representam gráficos complementares (símbolos arquitetônicos, gestos comunicativos e anotações). Esta funcionalidade pode ser focada principalmente no reconhecimento de geometria em 2D, já que é a mais utilizada entre arquitetos. A variação de pressão registrada na aquisição pode ser utilizada para ajudar na discriminação de formas e para atribuir diferentes tipos de linhas. A intervenção do usuário neste processo pode ser incluída para aproveitar sua inteligência visual para reconhecer formas emergentes;

(iii) **Interpretação arquitetônica:** computar a geometria reconhecida como o elemento arquitetônico em 3D pretendido pelo usuário no seu contexto (volumetria, pilares, paredes, janelas, etc.) e as relações espaciais entre eles e com o entorno (localização), baseado na geometria de edificação e em gráficos complementares.

(iv) **Reconstrução:** quando a interpretação provê de suficiente contexto, o modelo 3D deve ser gerado através de processos de reconstrução automática, basicamente a partir de geometria arquitetônica em 2D, seguindo as regras da linguagem gráfica do arquiteto. O processo pode ser complementado com reconstrução gestual para lidar com ambiguidades, como por exemplo, a inclusão de entradas do usuário para decidir e confirmar possíveis reconstruções, ou para

definir parâmetros não explicitados, como alturas numa terceira dimensão não desenhada. Como alternativa, uma base de dados de conhecimento sobre as relações mais comuns entre elementos arquitetônicos pode ser usada para ajudar na interpretação, mas só se for implementada de uma maneira que não limite as possibilidades de definir soluções arquitetônicas inovadoras.

4.1.3 Simulações preliminares de desempenho

Modificar temporariamente a aparência do modelo para representar o desempenho preliminar da edificação sendo projetada. O software deve exibir uma representação não fotorealística do projeto sendo desenvolvido por modelagem baseada em sketch, e substituir esta visualização quando valores de desempenho estejam disponíveis. Em momentos de inatividade, ou por requisição do usuário, o software deveria executar análises de desempenho e exibir os resultados diretamente no modelo, amostrando-o com uma malha poligonal de densidade constante, na qual os vértices podem definir cores simbólicas (p.ex., representando valores de radiação solar) ou serem movidos (p.ex., representando deformações estruturais) simulando as consequências físicas para cada vértice. Os distintos fatores de desempenho requerem diferentes recursos computacionais para analisar e visualizar resultados de desempenho, mas as simulações preliminares de desempenho devem ser aproximações simplificadas, não só para garantir uma rápida execução, mas para responder adequadamente aos problemas mal definidos das etapas iniciais de projeto.

4.1.4 Transição entre etapas de projeto

Os formatos de arquivos devem atender padrões da indústria da construção civil ou serem compatíveis com estes para oferecer interoperabilidade com os softwares mais usados nas etapas avançadas de projeto arquitetônico, permitindo uma fácil transição. Para abordar a interoperabilidade com sistemas BIM, o formato IFC pode ser utilizado. Uma abordagem mais simples poderia considerar a interoperabilidade com os softwares que são relevantes só nas etapas imediatamente subsequentes às iniciais. Em tal abordagem, os padrões *de facto* da indústria seriam os formatos DWG ou DXF (AutoCAD) e SKP (SketchUp). O suporte a estes formatos é a chave para permitir fluxo bidirecional de modificações de projeto.

No decorrer do desenvolvimento desta investigação, é notória a importância da aplicação que um software como imaginado nos cenários de uso tem na evolução das técnicas de projeto de arquitetura e urbanismo. Na medida em que avançamos na esfera tecnológica, a arquitetura não pode mais ser restringida ao ensaio ou rabisco no plano 2D. O processo de

projeto deve ter a possibilidade de acompanhar a mente que o imagina e desenvolve, tendo a possibilidade de acompanhar os insights de sua criação.

4.2 DESENVOLVIMENTO FUTURO

A natural continuação desta pesquisa é a implementação das funcionalidades destacadas nesta dissertação, num aplicativo de software para usuários finais, seguindo os critérios descritos. No entanto, além do trabalho técnico de implementação, existem alguns temas que necessitarão ser aprofundados, como:

- Aprofundar o estudo sobre o reconhecimento de sketches, diferenciando sketches de pensamento e sketches prescritivos. Considerar, para sketches de pensamento, o reconhecimento de imagens por meio de vetorização e análise geométrica como metodologias para conduzir a emergência de formas, assim como em Gero e Yan (1993), Terzidis (1998), Gero e Saunders (2000), e Jupp e Gero (2003)
- Aprofundar o estudo sobre interpretação de formas no contexto arquitetônico e urbano: discriminação de elementos de diferentes classes (ex.: ruas, quadras, lotes, edifícios, etc.), considerando o trabalho adiantado em (2009) Turkienicz, Gonçalves e Grazziotin (2008).
- Estudo mais detalhado de quais fatores de desempenho são mais apropriados de simular nas etapas iniciais e qual tipo de computação e representação é mais vantajosa para estas simulações.
- Avaliar diferentes serviços web de mapas interativos para a melhor integração no aplicativo, para a definição de geo-localização, orientação e escala do projeto.
- Avaliação da interação e testes de usabilidade com usuários finais.

Adicionalmente, a reflexão sobre esta pesquisa levanta mais assuntos que podem bifurcar para outros projetos de pesquisa, entre os que podemos nomear:

- Funcionalidades de projeção colaborativa, considerando o reconhecimento de sketches comunicativos, como sugerido por Company, Contero *et al.* (2006), com vocabulário de signos relacionado a desenhos gestuais baseados no observado por Suwa, Purcell e Gero (1998).
- Captura de áudio e vídeo de webcams, permitindo processos de projeção *think-aloud* e interação por gestos no ar para visualização em realidade aumentada, como em Trevisan, Carvalho *et al.* (2010) e em Araújo, Jorge e Duarte (2012).

- Desenho paramétrico de sketches, definindo restrições geométricas a partir de signos, assim como em Naya, Contero *et al.* (2008).
- Estudar a possibilidade de desenvolvimento destas funcionalidades num *plug-in* que funcione dentro de um software CAD ou BIM de uso representativo nas etapas avançadas de projeto.

É visível que há trabalho a ser feito para o desenvolvimento deste aplicativo e muitas possibilidades de avanço a partir daí. Também é destacável que este trabalho traz de volta à discussão tópicos clássicos do campo da computação, como visão computacional e computação baseada em conhecimento, reafirmando estas visões não excludentes dentro do campo do design assistido por computador, como antecipado pelos pioneiros da computação gráfica.

5 REFERÊNCIAS

- ALAÇAM, S.; ÇAĞDAŞ, G. **An Interface Proposal for Collaborative Architectural Design Process**. Architecture in Computro: 26th eCAADe Conference Proceedings. Antwerpen (Belgium): The Higher Institute of Architectural Sciences, Henry van de Velde. 2008. p. 319-324.
- ANDERSON, J.; SCHWAGER, P.; KERNS, R. The Drivers for Acceptance of Tablet PCs by Faculty in a College of Business. **Journal of Information Systems Education**, 17, n. 4, 2006. 429-440.
- AOSP – Android Open Source Project. Android Design Principles. In: **Android Developers**, Disponível em: < <http://developer.android.com/design/get-started/principles.html>>. Acesso em: 20 jan. 2014.
- ANSARI, S. Autodesk Releases SketchBook Mobile on the App Store. **PR Newswire**, 2009. Disponível em: <<http://www.prnewswire.com/news-releases/autodesk-releases-sketchbook-mobile-on-the-app-store-62199897.html>>. Acesso em: 5 mai 2014.
- APPLE INC. **iOS Human Interface Guidelines**. Disponível em: < <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/>>. Acesso em 15 nov. 2014
- ARAÚJO, B.; JORGE, J.; DUARTE, J. Combining Virtual Environments and Direct Manipulation for Architectural Modeling. **Proceedings of eCAADe 2012**, Prague, 2012. 419-428.
- ASANOWICZ, A. **Form Follows Media: Experiences of Bialystok**. AVOCAAD Conference Proceedings. Brussels: DUP Science - Delft University Press. 2003. p. 81-90.
- BEAUDOUIN-LAFON, M. **Designing Interaction, not Interfaces**. Proceedings of the Working Conference on Advanced Visual Interfaces. New York: The ACM Press. 2004. p. 15-22.
- BEAUDOUIN-LAFON, M.; MACKAY, W. Prototyping tools and techniques. In: SEARS, A.; JACKO, J. **The human-computer interaction: Development Process - Human Factors and Ergonomics**. [S.l.]: L. Erlbaum Associates Inc., 2002. p. 1006-1031. ISBN 9781420088892.
- BHAVNANI, S. et al. CAD Usage in an Architectural Office: From Observations to Active Assistance. **Automation in Construction**, 5, n. 3, 1996. 243-255.
- BHAVNANI, S.; GARRETT, J.; SHAW, D. **Leading Indicators of CAD Experience**. CAAD Futures. [S.l.]: Elsevier. 1993. p. 313-334.
- BHAVNANI, S.; JOHN, B. The Strategic Use of Complex Computer Systems. In: CARROLL, J. **Human-Computer Interaction in the New Millennium**. Boston: Addison-Wesley, 2001. Cap. 5, p. 97-124. ISBN 0201704471.
- BRANDL, P. et al. **Combining and Measuring the Benefits of Bimanual Pen and Direct-Touch Interaction on Horizontal Interfaces**. Proceedings of the working conference on Advanced Visual Interfaces. New York: ACM Press. 2008. p. 154-161.
- BROOKS, R.; ROSENBERG, C. **L - a Common Lisp for embedded systems**. Association of Lisp Users Meeting and Workshop LUV. [S.l.]: [s.n.]. 1995.
- BRUCE, M.; BESSANT, J. **Design in Business: Strategic Innovation Through Design**. Harlow: Pearson Education Limited, 2002.

- BÜRDEK, B. **Design: History, Theory and Practice of Product Design**. Berlin: Birkhäuser Publishers for Architecture, 2005. 479 p.
- BUXTON, W. **Chunking and Phrasing and the Design of Human-Computer Dialogues**. Information Processing 86: Proceedings of the IFIP 10th World Computer Congress. Dublin: North-Holland/IFIP. 1986. p. 475-480.
- BUXTON, W. The Long Nose of Innovation. **Businessweek**, Nova York, n. (Innovation & Design), 2 ene 2008. Disponível em: <<http://www.businessweek.com/stories/2008-01-02/the-long-nose-of-innovationbusinessweek-business-news-stock-market-and-financial-advice>>. Acesso em: 5 feb 2014.
- BUXTON, W. Multi-touch Systems that I Have Known and Loved. **Bill Buxton Home Page**, 19 mar 2013. Disponível em: <<http://www.billbuxton.com/multitouchOverview.html>>. Acesso em: 30 nov 2013.
- CARLSON, W. CAD/CAM/CADD/CAE software defines a new field. **A Critical History of Computer Graphics and Animation**, 2003. Disponível em: <<http://design.osu.edu/carlson/history/lesson10.html>>. Acesso em: 20 nov 2013. Section 10.
- CARPO, M. The Art of Drawing. **Architectural Design**, v. 83, n. 5, p. 128–133, 2013.
- CHENG, N.; KIEFERLE, J. More than Skin deep: Early Design Evaluation Tools. **International Journal of Architectural Computing**, Liverpool, 6, n. 1, 2008. iii-iv.
- CHRABIN, A.; NEUCKERMANS, H.; SZEWCZYK, J. **A Critical Evaluation of Early Stages Software in its Capacity of Coping with Contextual Issues**. Local Values in a Networked Design World - Added Value of Computer Aided Architectural Design. [S.l.]: DUP Science - Delft University Press. 2003.
- COHN, D. Evolution of Computer-Aided Design: How we got to where we are, and where are we headed. **Desktop Engineering**, 1 dec 2010.
- COMPANY, P. et al. **Aplicación docente de un programa de modelado 3D mediante bocetos axonométricos**. XIII ADM – XV INGEGRAF International Conference on Tools and Methods Evolution in Engineering Design. Napoli: ADM. 2003.
- COMPANY, P. et al. **A study of usability of sketching tools aimed at supporting prescriptive sketches**. Proceedings of the Third Eurographics conference on Sketch-Based Interfaces and Modeling. Aire-la-Ville (Switzerland): Eurographics Association. 2006. p. 139-146.
- COMPANY, P.; PIQUER, A.; CONTERO, M. **On the evolution of geometrical reconstruction as a core technology to sketch-based modeling**. Proceedings of Eurographics Workshop on Sketch-Based Interfaces and Modeling. Grenoble (France): The Eurographics Association. 2004.
- COMPUTER Sketchpad (Science Reporter). Direção: R Morash. Produção: R Morash. [S.l.]: Massachusetts Institute of Technology. 1963. Som.
- CONTERO, M. et al. **CIGRO: A Minimal Instruction Set Calligraphic Interface for Sketch-Based Modeling**. International Conference on Computational Science and Its Applications. Montreal: [s.n.]. 2003. p. 549-558.
- CONTERO, M. et al. **Improving visualization skills in engineering education**. IEEE Computer Graphics and Applications. [S.l.]: [s.n.]. 2005. p. 24–31.

- CRAWLEY, D. et al. Contrasting the capabilities of building energy performance simulation programs. **Building and Environment**, 43, n. 4, 2008. 661-673.
- CZELUSNIAK, D. et al. Limitações das Atuais Soluções em Software: Uma Visão para Desenvolvimento de Ferramentas em Gestão de Projetos. **Revista Eletrônica de Sistemas de Informação**, 4, n. 1, 2009.
- DENZER, A.; GARDZELEWSKI, J. **Drawing and Modeling: Analog Tools in the Age of BIM**. Proceedings of the 2011 Architectural Engineering Conference. Oakland (CA): American Society of Civil Engineers. 2011. p. 44-53.
- DICHERRI, R. Autodesk SketchBook. **deviantART**, 2013. Disponível em: <<http://autodesk-sketchbook.deviantart.com/>>. Acesso em: 5 mai 2014. (Official Autodesk SketchBook Users Group on deviantART).
- DIX, A. et al. **Human-Computer Interaction**. 3. ed. New Jersey: Prentice Hall, 2003.
- DO, E. **The right tool at the right time: investigation of freehand drawing as an interface to knowledge based design tools**. Georgia Institute of Technology. [S.l.]. 1999.
- DORSEY, J. et al. **The mental canvas: A tool for conceptual architectural design and analysis**. 15th Pacific Conference on Computer Graphics and Applications, PG'07. [S.l.]: IEEE. 2007. p. 201-210.
- EASTMAN, C. **Cognitive process and ill-defined problems: A case study from design**. Proceedings of the 1st International Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers. 1969. p. 669-690.
- EL-GAYAR, O.; MORAN, M. **College students' acceptance of Tablet PCs: An application of the UTAUT Model**. 2006 Annual Meeting of the Decision Sciences Institute. San Antonio (TX): [s.n.]. 2006. p. 2845-2850.
- ELSEN, C. et al. Sketch-based interfaces for modeling and users' needs: Redefining connections. In: _____ **Artificial Intelligence for Engineering Design, Analysis and Manufacturing**. Cambridge (MA): Cambridge University Press, v. 26, 2012. p. 281-301. Special Issue 03.
- ELSEN, C.; DARSES, F.; LECLERCQ, P. **An anthropo-based standpoint on mediating objects: evolution and extension on industrial design practices**. Design Computing and Cognition '10. [S.l.]: Springer. 2011. p. 55-74.
- ELSEN, C.; LECLERCQ, P. "SketSha" - The Sketch Power to Support Collaborative Design. In: LUO, Y. **Cooperative Design, Visualization, and Engineering**. Berlin: Springer, 2008. p. 20-27. Lecture Notes in Computer Science, v. 5220.
- FASTE, T.; FASTE, H. **Demystifying "Design Research": Design is Not Research, Research is Design**. Proceedings of the Industrial Designers Society of America Education Symposium. Boston: [s.n.]. 2012.
- FIGUEREDO, R.; VIZIOLI, S. **Estudo Comparativo entre as Novas Ferramentas Digitais Utilizadas no Desenho à Mão Livre**. XVII Congreso de la Sociedad Iberoamericana de Gráfica Digital. Valparaiso: Universidad Técnica Federico Santa María. 2013. p. 87-91.
- FLORIO, W. Croquis de concepção no processo de projeto em Arquitetura. **Exacta**, São Paulo, 8, n. 3, 2010. 373-383.

- FONSECA, M.; JORGE, J. Experimental evaluation of an on-line scribble recognizer. **Pattern Recognition Letters**, 22, n. 12, oct 2001. 1311-1319.
- FORBUS, K. et al. **CogSketch**: Open-domain sketch understanding for cognitive science research and for education. EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling. [S.l.]: [s.n.]. 2008.
- FORLINES, C. et al. **Direct-Touch vs. Mouse Input for Tabletop Displays**. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York: ACM Press. 2007. p. 647-656.
- FRANCO, J. 2 en 1: La tecnología en computación que busca seguirle el paso a los arquitectos. **ArchDaily**, 23 jun 2014. Disponível em: <<http://www.archdaily.mx/325658>>. Acesso em: 24 jun 2014.
- GARDNER, H. **Frames of mind**: The theory of multiple intelligences. New York: Basic books, 1985.
- GERO, J. Design Prototypes: A Knowledge Representation Schema for Design. **AI Magazine**, v. 11, n. 4, p. 26-36, 1990.
- GERO, J.; SAUNDERS, R. **Constructed Representations and their Functions in Computational Models of Designing**. Proceedings of the Fifth Conference on Computer Aided Architectural Design Research in Asia. Singapore: CASA. 2000. p. 215–224.
- GERO, J.; YAN, M. **Discovering Emergent Shapes using a Data-Driven Symbolic Model**. CAAD Futures. [S.l.]: [s.n.]. 1993.
- GRABNER, T.; FRICK, U. GECO: Architectural Design Through Environmental Feedback. **Architectural Design**, Londres, 83, n. 2, 2 Apr 2013. 142-143. (Special Issue: Computation Works: The Building of Algorithmic Thought).
- GREENBERG, S; CARPENDALE, S; MARQUARDT, N; BUXTON, W. **Sketching User Experiences: The Workbook**. Boston: Morgan Kaufmann, 2011.
- GUIARD, Y. Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model. **Journal of Motor Behavior**, 19, n. 4, 1987. 486-517.
- HAMBLEN, M. Update: Galaxy Tab sales hit 1M mark. **Computerworld**, 5 Dec 2010. Disponível em: <http://www.computerworld.com/s/article/9199678/Update_Galaxy_Tab_sales_hit_1M_mark>. Acesso em: 5 Feb 2014.
- HEIM, S. **The Resonant Interface**: HCI Foundations for Interaction Design. New York: Pearson Addison-Wesley, 2008.
- HEROT, C.; WEINZAPFEL, G. **One-Point Touch Input of Vector Information for Computer Displays**. Proceedings of the 5th annual conference on Computer graphics and interactive techniques. Atlanta (GA): ACM. 1978. p. 210-216.
- HINCKLEY, K. et al. **Manual Deskterity**: An Exploration of Simultaneous Pen + Touch Direct Input. CHI '10 Extended Abstracts on Human Factors in Computing Systems. New York: ACM. 2010a. p. 2793-2802.
- HINCKLEY, K. et al. **Pen + Touch = New Tools**. Proceedings of the 23rd annual ACM symposium on User interface software and technology. New York: ACM. 2010b. p. 27-36.

HOFFMAN, D. **Visual Intelligence: How we create what we see.** [S.l.]: W.W. Norton & Company, 1998.

IBRAHIM, R.; RAHIMIAN, F. Comparison of CAD and manual sketching tools for teaching architectural design. **Automation in Construction**, 19, n. 8, 2010. 978-987.

INGRAHAM, N. Apple has sold 170 million iPads since it launched in April 2010. **The Verge**, 22 October 2013. Disponivel em: <<http://www.theverge.com/2013/10/22/4866966/apple-has-sold-170-million-ipads-since-it-launched-in-april-2010>>. Acesso em: 4 fev 2014.

JOHANSSON, R. Forget WYSIWYG editors - use WYSIWYM instead. **456 Berea Street**, 2006. Disponivel em: <http://www.456bereastreet.com/archive/200612/forget_wysiwyg_editors_use_wysiwym_instead/>. Acesso em: 16 jun 2014.

JUCHMES, R.; LECLERCQ, P.; AZAR, S. A freehand-sketch environment for architectural design supported by a multi-agent system. **Computers & Graphics**, 29, n. 6, 2005. 905-915.

JUPP, J.; GERO, J. **Towards computational analysis of style in architectural design.** IJCAI03 Workshop on Computational Approaches to Style Analysis and Synthesis. Acapulco: IJCAI. 2003.

KALLIO, K. **3D6B Editor: Projective 3D Sketching with Line-Based Rendering.** Proceedings of the 2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling. Dublin: Eurographics Association. 2005. p. 73-79.

KANG, Y. et al. Feature-based 3D CAD Modeling on Smart Device Using Multi-touch Gesture. **International Journal of CAD/CAM**, 13, n. 2, 2013. 49-62.

KELLY, N.; GERO, J. **Constructive Interpretation in Design Thinking.** Proceedings of the 27th Conference on Education and Research in Computer Aided Architectural Design in Europe. Istanbul: Istanbul Technical University and Yildiz Technical University. 2009. p. 97-104.

KIM, D.; KIM, M. A new modeling interface for the pen-input displays. **Computer-Aided Design**, 38, n. 3, 2006. 210-223.

KRUEGER, M. **Artificial Reality.** Reading (MA): Addison-Wesley, 1983.

KRUEGER, M.; GIONFRIDDO, T.; HINRICHSEN, K. **VIDEOPLACE - An Artificial Reality.** Proceedings of the ACM Conference on Human Factors in Computing Systems. [S.l.]: ACM. 1985. p. 35-40.

LAWSON, B. **What Designers Know?** Oxford (UK): Elsevier Architectural Press, 2004.

LAWSON, B. **How Designers Think: The Design Process Demystified.** Oxford (UK): Elsevier Architectural Press, 2005.

LECLERCQ, P. Invisible sketch interface in architectural engineering. In: _____ **Lecture notes in computer science.** Berlin: Springer, v. 3088, 2004. p. 353-363.

LEGANCHUK, A.; ZHAI, S.; BUXTON, W. Manual and Cognitive Benefits of Two-Handed Input: An Experimental Study. **Transactions on Human-Computer Interaction**, 5, n. 4, 1998. 326-359.

LEGASPI, C. How to use Photoshop Brush Settings. **Freshdesigner**, 20 jul 2010. Disponivel em: <<http://www.freshdesigner.com/how-to-use-photoshop-brush-settings/>>. Acesso em: 5 mai 2014.

- LONG, A. **Quill: A Gesture Design Tool for Pen-based User Interfaces**. University of California at Berkely. Berkely. 2001. Thesis (PhD in Computer Science).
- LOPES, P. et al. **Bimanual manipulation and pen-based input for 3D modelling**. Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling. Annecy (France): Eurographics Association. 2011. p. 15-22.
- MASRY, M.; LIPSON, H. **A Sketch-Based Interface for Iterative Design and Analysis of 3D Objects**. Proceedings of the 2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling. Dublin: Eurographics Association. 2005. p. 109-118.
- MATTHEWS, C. **Case Studies in Engineering Design**. [S.l.]: Arnold, 1998.
- MEHTA, N. **A Flexible Machine Interface**. University of Toronto. Toronto. 1982.
- MITCHELL, W.; MCCULLOUGH, M. **Digital Design Media**. 2ª. ed. Ithaca: Harvard University Press, 1995.
- MONEDERO, J. **Aplicaciones Informáticas en Arquitectura**. 2ª. ed. Barcelona: Ediciones UPC, 2000.
- MOREIRA, D.; KOWALTOWSKI, D. Discussão sobre a importância do programa de necessidades no processo de projeto em arquitetura. **Ambiente Construído**, Porto Alegre, 9, n. 2, 2009. 31-45.
- MURUGAPPAN, S.; RAMANI, K. **FEAsy: A Sketch-Based Interface Integrating Structural Analysis in Early Design**. ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. San Diego: ASME. 2009. p. 743-752.
- MUSIL, S. Microsoft renames its old Surface computer as PixelSense. **CNET**, 18 jun. 2012. Disponível em: <www.cnet.com/news/microsoft-renames-its-old-surface-computer-as-pixelsense/>. Acesso em: 02 abr. 2014.
- NAKATANI, L.; ROHRLICH, J. **Soft Machines: A Philosophy of User-Computer Interface Design**. Proceedings of the ACM Conference on Human Factors in Computing Systems. [S.l.]: ACM. 1983. p. 12-15.
- NAYA, F. et al. **Smart Sketch System for 3D Reconstruction Based Modeling**. International Symposium on Smart Graphics. Heidelberg: Springer-Verlag. 2003. p. 58–68.
- NAYA, F. et al. Parametric freehand sketches. **Lecture Notes in Computer Science**, 3044, 2004. 114-120.
- NAYA, F. et al. Sketch-Based Interfaces for Parametric Modelling. In: PAVLIDIS, I. **Human Computer Interaction**. [S.l.]: InTech, 2008. p. 44-55.
- NEGULESCU, M.; RUIZ, J.; LANK, E. **Exploring Usability and Learnability of Mode Inferencing in Pen/Tablet Interfaces**. Eurographics Workshop on Sketch-Based Interfaces and Modeling. [S.l.]: Eurographics Association. 2010. p. 87-94.
- NICKOLS, F. The Knowledge in Knowledge Management. In: CORTADA, J.; WOODS, J. **The Knowledge Management Yearbook 2001-2002**. [S.l.]: Butterworth-Heinemann, 2002.
- NOME, C.; QUEIROZ, N. **Performance do Edifício: Produtos e Processo para Integração no Projeto Arquitetônico**. Proceedings of the 17th Conference of the Iberoamerican Society of Digital Graphics. Valparaíso: [s.n.]. 2013. p. 357-361.

- OH, J.; STUERZLINGER, W.; DANAHY, J. **Sesame**: towards better 3d conceptual design systems. Proceedings of the 6th conference on Designing Interactive systems. New York: The ACM Press. 2006. p. 80-89.
- OLSEN, L. et al. Sketch-based modeling: A survey. **Computers & Graphics**, 33, n. 1, feb 2009. 85-103.
- OXMAN, R. Design by re-representation: a model of visual reasoning in design. **Design Studies**, Milton Keynes (UK), 18, n. 4, oct 1997. 329-347.
- OXMAN, R. The thinking eye: visual re-cognition in design emergence. **Design Studies**, Milton Keynes (UK), 23, n. 2, mar 2002.
- OXMAN, R. Performative design: a performance-based model of digital architectural. **Environment and Planning B: Planning & Design**, 36, n. 6, 2009. 1026-1037.
- PRANOVICH, S. **Structural Sketcher: A Tool for Supporting Architects in Early Stages**. Technische Universiteit Eindhoven. Eindhoven, p. 150. 2004.
- RAYMOND, E. YAFIYGI. **The Jargon File**, 1995. Disponível em: <<http://catb.org/~esr/jargon/html/Y/YAFIYGI.html>>. Acesso em: 16 jun. 2014.
- REAS, C.; MCWILLIAMS, C.; BARENDSE, J. **Form+code in design, art, and architecture**. New York: Princeton Architectural Press, 2010. 176 p. (Design briefs).
- REBELO, I. **Interação Entre Homem e Computador**. Centro Universitário Unieuro. Brasília. 2009.
- RIGHI, T.; CELANI, G. **Tecnologias de displays interativos no processo de projeto arquitetônico**. Proceedings of the 12th Iberoamerican Congress of Digital Graphics. La Habana: CUJAE. 2008. p. 285-289.
- RUTTEN, D. Galapagos: On the logic and limitations of generic solvers. **Architectural Design**, Londres, 83, n. 2, 2 Apr 2013. 132-135. (Special Issue: Computation Works: The Building of Algorithmic Thought).
- SAORÍN, J. et al. Tabletas digitales para la docencia del dibujo, diseño y artes plásticas. **Teoría de la Educación: Educación y Cultura en la Sociedad de la Información**, Salamanca, v. 12, n. 2, 2011. p. 259-279.
- SCHMIDT, R. et al. **Shapeshop**: Sketch-based solid modeling with blobtrees. 2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling. Dublin: Eurographics Association. 2005. p. 53-62.
- SHERWOOD, B.; STIFLE, J. **The PLATO IV Communications System**. University of Illinois. Urbana, p. 33. 1975.
- SKALAK, D. **Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms**. ICML. [S.l.]: [s.n.]. 1994. p. 293-301.
- SPACECLAIM CORP. **SpaceClaim Showcases Multi-touch for 3D Engineering Design**. Concord (MA). 2009.
- STEELE, J. **Arquitectura y revolución digital**. Tradução de J. Escofet. 1ª. ed. Barcelona: Gustavo Gili, 2001. 240 p.

STILL, J.; DARK, V. Cognitively describing and designing affordances. **Design Studies**, Milton Keynes (UK), 34, n. 3, 2013. 285-301.

SUTHERLAND, I. **Sketchpad: a man-machine graphical communication system**. Massachusetts Institute of Technology. Cambridge (EUA), p. 183 f. 1963.

SUWA, M. Constructive perception: Coordinating perception and conception toward acts of problem-finding in a creative experience. **Japanese Psychological Research**, 45, n. 4, nov 2003. 221-234.

SUWA, M.; PURCELL, T.; GERO, J. Macroscopic analysis of design processes based on a scheme for coding designers' cognitive actions. **Design Studies**, Milton Keynes (UK), 19, n. 4, oct 1998. 455-483.

SUWA, M.; TVERSKY, B. What architects and students perceive in their sketches: A protocol analysis. **Design Studies**, Milton Keynes (UK), 18, n. 4, oct 1997. 385-403.

TAMASHIRO, H. **Desenho técnico arquitetônico: constatação do atual ensino nas escolas brasileiras de arquitetura e urbanismo**. Universidade de São Paulo. São Carlos. 2003. Dissertação (Mestrado em Tecnologia do Ambiente Construído).

TERZIDIS, K. **A System for Extracting Aesthetic Information from Architectural Drawings**. MVA '98 LAPR Workshop on Machine Vision Applications. Chiba (Japan): Makuhan. 1998. p. 73-76.

TREVISAN, D. et al. **Supporting the design of multimodal interactions: a case study in a 3D sculpture application**. In: XII Symposium on Virtual and Augmented Reality. Natal: [s.n.], 2010.

TRIMBLE NAVIGATION LTD. **SketchUp | 3D for Everyone**, 2013. Disponível em: <www.sketchup.com>. Acesso em: 1 dec. 2013.

TURKIENICZ, B.; GONÇALVES, B.; GRAZZIOTIN, P. CityZoom: A Visualization Tool for the Assessment of Planning Regulations. **Internationa Journal of Architectural Computing**, Liverpool, 6, n. 1, 2008. 79-95.

VANNINI, V. **A otimização da forma para captação de radiação solar sobre superfícies de edifícios: um exercício de integração entre os programas Rhinoceros e Ecotect**. Universidade Federal do Rio Grande do Sul. Porto Alegre. 2011. Dissertação (Mestrado em Arquitetura).

VANNINI, V.; BUENO, E.; TURKIENICZ, B. **Otimização geométrica de superfícies de fachada para uso fotovoltaico**. Forma (In)Formação: Anais do XVI Congresso da Sociedade Iberoamericana de Gráfica Digital. Fortaleza: UFC/UNIFOR. 2012. p. 426-430.

VILLAMOR, C.; WILLIS, D.; WROBLEWSKI, L. Touch Gesture Reference Guide. **LukeW Ideation + Design**, 2010. Disponível em: <<http://static.lukew.com/TouchGestureGuide.pdf>>. Acesso em: 15 mai 2014.

VIZIOLI, S.; LANCHI, J.; CASTRAL, P. **Drawing and Surface: Freehand Architectural Drawing Mediated by Digital**. Electronic Visualisation and the Arts. London: EVA. 2013. p. 174-180.

WANG, H.; WANG, C. Open source software adoption: A status report. **IEEE Software**, v. 18, n. 2, p. 90-95, 2001.

WEINZAPFEL, G.; JOHNSON, T.; PERKINS, J. **IMAGE: An interactive computer system for multi-constrained spatial synthesis**. DAC 1971. [S.l.]: [s.n.]. 1971. p. 101-108.

WINSTON, A. Survey reveals industry split over BIM. **Building Design**, 7 dec. 2010. Disponível em: <<http://www.bdonline.co.uk/news/survey-reveals-industry-split-over-bim/5010009.article>>. Acesso em: 4 nov. 2013.

WOO, J. et al. Dynamic Knowledge Map: reusing experts' tacit knowledge in the AEC industry. **Automation in Construction**, 13, n. 2, Mar 2004. 203-207.

YEE, R. **Desenho Arquitetônico: um compêndio visual de tipos e métodos**. Tradução de L Silva. 3ª. ed. Rio de Janeiro: LTC, 2009.

YU, J.; ZHANG, K. **A Prototype Sketch-Based Architectural Design System with Behavior Mode**. Simon Fraser University. Burnaby (Canada), p. 7. 2007.

GLOSSÁRIO

AEC: siglas de *architecture, engineering and construction*, ou arquitetura, engenharia e construção, comumente usadas na informática para se referir à indústria ou área de atuação de alguns softwares CAD, de modelagem em 3D e BIM. Também se usa o acrônimo AECO.

Affordance: potencial dos objetos de sugerir intuitivamente os seus possíveis usos a partir da percepção das características destes objetos. Em IHC, é a propriedade de interação de um dispositivo que define a percepção das suas funcionalidades.

BIM: acrônimo de *building information modeling*, modelagem da informação de construção, sistema de aquisição e gestão de informações AEC sobre um projeto de edificação a partir de um modelo 3D vinculado a uma base de dados. Método de trabalho atribuído às etapas avançadas de projeto, gestão de obra e monitoramento do ciclo de vida de uma edificação.

Bimanual: Veja-se **interação bimanual**.

CAD: acrônimo de *computer-aided design*, projeto assistido por computador. Em arquitetura, o termo é comumente atribuído ao paradigma de modelagem ao que pertencem os softwares de desenho técnico, como AutoCAD. Em engenharia mecânica, alguns autores também usam o termo para incluir softwares de MCAD. A relação do CAD com o apoio a atividades de projeto mais aprofundadas do que o desenho está atualmente em desuso. Alguns sugerem a revisão da significância do acrônimo para *computer-aided drafting*, desenho assistido por computador.

CHI: acrônimo de *computer-human interface*, ou interface computador-humano. Antigo termo usado para denominar o atual campo da **IHC**. Ver os termos **interface** e **interação**.

Funcionalidade: conjunto de funções ou características de uma aplicação de software oferecidas para apoiar as ações do usuário dentro das atividades de uma área de atuação.

GUI: acrônimo de *graphical user interface*, interface gráfica de usuário. É uma interface virtual que permite a interação entre um programa de computador e um usuário, a partir da renderização, num monitor, de elementos gráficos interativos, como ponteiros ou botões.

IHC: siglas de interação humano-computador. Campo interdisciplinar que estuda o design, a engenharia e a psicologia das interações entre usuários e sistemas de computador. Usado também na sua forma inglesa *HCI (human-computer interaction)*. Também conhecido como interação homem-máquina. O termo substitui ao **CHI**.

Interação humano-computador: comunicação e influência mútua entre usuários e sistemas computacionais através de protocolos de uso em **interfaces de usuário**, onde acontecem

ações num ou noutro sentido, para a ativação das funcionalidades oferecidas pelo sistema computacional.

Interação bimanual: modo de interação humano-computador na qual o usuário utiliza as duas mãos para controlar dispositivos de entrada. Tradicionalmente referido ao modo de trabalho com mouse na mão dominante e teclado na mão não dominante, na inserção de atalhos de teclado em softwares CAD; mas também inclui experimentações, como dois mouses, mouse em combinação com caneta *stylus*, etc.

Interação multimodal: interação que combina simultaneamente mais de um modo de interação, geralmente apoiado na inserção bimanual. A mais relevante combina entrada por caneta *stylus* na mão dominante e o toque dos dedos da mão não dominante, sobre uma tela *multi-touch* capaz de discernir entre os dois modos, usando um software que possa processar esta combinação.

Interface: ponto de interação entre uma máquina e outra entidade. Na computação, a **interface de usuário** é a parte de um sistema computacional que permite que o usuário possa interagir com este sistema. Esta pode ser uma interface física (ex.: botões plásticos) ou gráfica (GUI). Na história da computação, o campo de estudos de interação era referido como de interface ou **CHI**.

Interpretação: análise semântica da informação reconhecida, dando-lhe significado, função, e/ou contexto (ex.: identificação de elementos arquitetônicos representados em planta a partir de formas primitivas).

MCAD: termo que se refere a uma subcategoria dos softwares CAD especializada na modelagem de peças de engenharia mecânica, valendo-se de paradigmas como o de modelagem paramétrica baseada em características (*feature-based parametric modeling*) para estabelecer relações explícitas entre os parâmetros das peças (ex.: restrições geométricas). Entre eles estão CATIA, SolidWorks, Creo, Inventor, entre outros.

Multimodal: veja-se **interação multimodal**.

Multi-touch: sistema de interação humano-computador definido pela utilização de múltiplos toques sobre uma tela que capta o toque dos dedos simultaneamente, ou tela *multi-touch*.

Reconhecimento: rotular, classificar dados adquiridos em bruto, para poder discriminar entre tipos de dados para futuros processamentos da informação (ex.: reconhecimento de traços de um sketch, identificando formas primitivas e elementos de texto).

Reconstrução: geração de um modelo 3D, cujas características contextuais relembram às sugeridas em representações mais abstratas, como em desenhos 2D, assim interpretadas

(ex.: geração do modelo 3D de um edifício, a partir de desenhos 2D de linhas, interpretadas como a representação isométrica de um edifício).

SBIM: siglas de *Sketch-Based Interfaces and Modeling*, título de uma série de seminários organizados pela Eurographics Association entre 2004 e 2011, onde se formalizaram os estudos interinstitucionais sobre modelagem baseada em sketch. Termo sugerido por Olsen, Samavati *et al.* (2009) para nomear a categoria de softwares que abordam este paradigma de modelagem, onde o modelo 3D é reconstruído a partir de sketches.

Simulação: representação computacional abstrata de um sistema físico para auxiliar o entendimento de um fenômeno ou processo. Neste trabalho, o termo é utilizado no contexto das simulações de desempenho de projetos de edificações, como uma representação do modelo projetado, ao qual se lhe atribuem aparências não realísticas, para visualizar o desempenho do projeto perante as suas características, como fatores ambientais: sombreamento, iluminância, etc.

Sketch: desenho em rascunho. Neste trabalho, refere-se ao desenho a mão livre elaborado como instrumento de apoio ao processo de projeto arquitetônico, conhecido nessa área como croquis. Sketch é o termo utilizado em Pesquisa em Design e nas Ciências da Computação.

Stylus: caneta digitalizadora. Dispositivo de entrada que se usa para desenhar, replicando o uso de uma caneta tradicional sobre uma mesa digitalizadora, *tablet* opaca ou tela sensível ao toque.

Tablet: qualificativo de dispositivo computacional em referência a suas características de entrada, como em **tablet opaca:** placa que funciona como dispositivo periférico de entrada, sobre a qual se desenha com uma caneta *stylus*; ou **computador tablet:** computador portátil que, diferentemente do laptop, possui funcionalidades de captação do toque na tela, fazendo desta um dispositivo de entrada direta embecida no computador.