

ARTIGO

PRUMO

Interação entre o arquiteto e as ferramentas digitais nas etapas iniciais de projeto¹

Ernesto Bueno

Doutorando da Universidade Presbiteriana Mackenzie-SP. Professor da Universidade Positivo, Curitiba-PR.

Contato: ernestobueno@gmail.com

¹Artigo desenvolvido a partir da dissertação de mestrado do autor no Programa de Pós-Graduação em Design da Universidade Federal do Rio Grande do Sul (PGDESIGN/UFRGS), sob orientação do professor Dr. Benamy Turkienicz. Parte do trabalho foi publicado em inglês no *International Journal of Architectural Computing* (BUENO; TURKIENICZ, 2014). Gostaria de agradecer ao Prof. Benamy pela valiosa orientação. Este trabalho foi realizado com apoio parcial da CAPES - Brasil, e com apoio parcial da FAURGS.

RESUMO

Nas etapas iniciais de projeto, o arquiteto trabalha aplicando conhecimento implícito para tomar decisões inicialmente difusas e progressivamente melhor definidas. Esta progressão precisa de representações que se adaptem a este processo reflexivo, como o croquis. Dentre as muitas tecnologias digitais disponíveis, os arquitetos não contam com uma que apoie desta maneira as etapas iniciais, trazendo problemas de fluxo de trabalho e deixando de abordar soluções de desempenho. Apresenta-se um estudo comparativo das funcionalidades de *software* experimentais dentre 2003 e 2014 que, utilizando hardware disponível, apoiam estas atividades; e descreve-se como estas funcionalidades devem ser integradas para atender às necessidades de desenho e projeto arquitetônico nas etapas iniciais.

Palavras-chave: Interação Humano-Computador; Projeto Arquitetônico; Modelagem Digital; Simulação de Desempenho

ABSTRACT

In the early stages of design, the architect works by applying implicit knowledge to make initially diffuse and progressively better-defined decisions. This progression needs a kind of representation that could adapt to this reflective process, such as the sketch. Among the many digital technologies available, architects do not count with one that supports the early stages of design in this way. This absence leads to workflow issues and fails to address performance solutions. A comparative study is presented, in which experimental *software* functionalities existing between 2003 and 2014 were tested using available hardware, to support these activities. Furthermore, it described how these functionalities should be integrated to meet the needs in architectural drawing and design in the early stages.

Key-words: Human-Computer Interaction; Architectural Design; Digital Modeling; Performance Simulation

Introdução

Avanços tecnológicos forçam mudanças metodológicas de projeto de arquitetura. Nas últimas décadas, estas mudanças acontecem em várias direções, com distintas motivações e diferentes consequências na produção arquitetônica. Dentre as vertentes tecnológicas que, apoiadas em novas tecnologias, têm maior presença, podemos mencionar brevemente três:

(I) Buscando potencializar a criatividade, otimizar o desempenho, viabilizando o desenho e execução de geometria complexa, temos tecnologias como a modelagem paramétrica, algorítmica e a fabricação digital, nas quais se apoia o design generativo. (II) Visando agilizar o desenvolvimento do projeto junto a sua documentação de execução de maneira fluida e interconectada entre diferentes atores da construção civil através das etapas de projeto e ciclo de vida da edificação, temos o BIM. Ainda, temos a tecnologia de realidade virtual (III), que tem sido introduzida na pesquisa em arquitetura e finalmente está tendo um uso representativo no exercício profissional em projetos e mais ainda na visualização destes, principalmente na interlocução com o cliente.

Porém, além da produção de arquitetos emblemáticos, alguns resultados de concursos e da presença nos meios de divulgação de projetos, estas mudanças ainda estão numa minoria do ofício do arquiteto. Pesquisas demonstram que a maioria dos arquitetos atuantes em projeto, o fazem aplicando metodologias estabelecidas no século XX. Estas metodologias definem uma progressão do projeto marcada basicamente por desenhos em quatro tipos principais de representação: croquis, modelo 3D, análise de desempenho e desenho técnico. Cada uma destas representações é tipicamente produzida com uma técnica e ferramenta apresentada na Tabela 1.

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

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

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

Nesta progressão típica, as ferramentas utilizadas pertencem a dois grandes grupos de mídias: a física (o papel) e as mídias digitais, separando as representações em dois domínios claramente definidos (Figura 1).

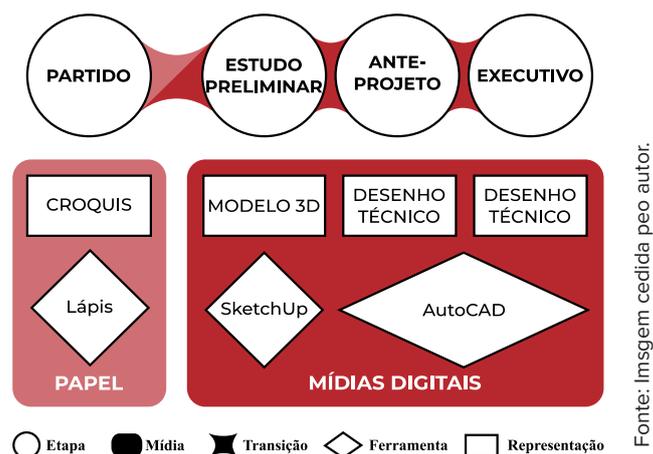


Figura 1: Transição de mídias no processo de projeto de arquitetura.

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 física para a digital. (YU; ZHANG, 2007)

Esta transferência é normalmente feita por meio da digitalização e redesenho em CAD, em *layers* ou camadas superpostas acima da imagem escaneada do

croquis (mapa de bits); ou através da construção de um novo desenho CAD, iniciado 'de zero' (sem uma prévia inserção de imagens), com base na observação e recriação das características do croquis (MASRY; LIPSON, 2005). Esta transferência consome um tempo de projeto no qual, pela interpretação, são revisadas decisões de projeto já tomadas (DENZER; GARDZELEWSKI, 2011) até que a representação atinja a definição apropriada para abordar os problemas das etapas mais avançadas de projeto (MOREIRA; 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)

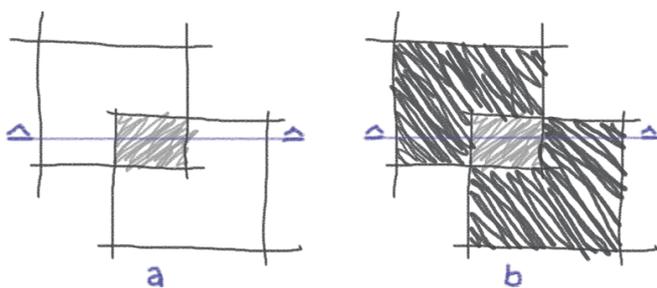
O ato de desenhar, como atividade projetual, é uma transação visual e mental, uma conversação que o projetista estabelece consigo mesmo (SCHÖN *apud* SUWA; TVERSKY, 1997) (OXMAN, 1997). O processo de ideação do projeto é conformado pelo ciclo reflexivo: desenhar, inspecionar, revisar. Os avanços de projeto são atingidos graças a seqüências 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 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 ma-

neira diferente à adotada num primeiro momento (GERO; YAN, 1993). A inteligência visual provê da habilidade de 'ver' coisas com a mente, interpretadas a partir das imagens inicialmente registradas através dos olhos (HOFFMAN, 1998). Esta inteligência é especialmente desenvolvida em arquitetos, que exploram este potencial com o treinamento visual dado pela experiência. Isto lhes habilita a encontrar formas e padrões emergentes que permitem a reinterpretar de um problema de projeto. O processo de identificação destas formas emergentes, ou **emergência visual**, faz explícitas as características e relações implícitas entre elementos originados por outras relações (GERO; YAN, 1993). Diferentemente do conceito de emergência tratado em outras áreas do conhecimento², 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 2 (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 2a), ou como um vazio no encontro entre dois volumes (Figura 2b). O projetista consegue interpretar estas formas emergentes como elementos arquitetônicos com diferentes propriedades espaciais que acontecem na terceira dimensão, ainda não representada. Na necessidade de externalizar estas propriedades, o projetista pode elaborar outros croquis que as ilustrem (Figura 3), 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 interpre-

tação 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.*).

Fonte: Imagem cedida pelo autor.

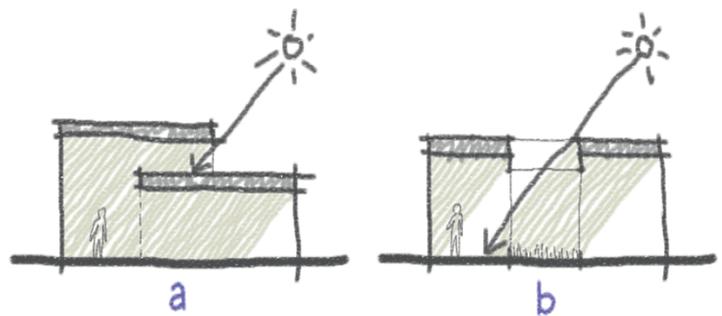


Problemas de interação em CAD

A indústria de *softwares* CAD tem sido grandemente influenciada pela evolução dos sistemas de interação dos computadores pessoais. O primeiro sistema comercial de *software* CAD, o AMD CATIA³, já contava com funcionalidades avançadas de desenho paramétrico (CARLSON, 2003). Mas a sua comple-

Figura 2: Reconhecimento de uma forma emergente num croquis arquitetônico em planta.

Figura 3: Croquis em corte que ilustram as consequências arquitetônicas de interpretações de uma forma emergente.



Fonte: Imagem cedida pelo autor.

Fonte: Imagem cedida pelo autor.

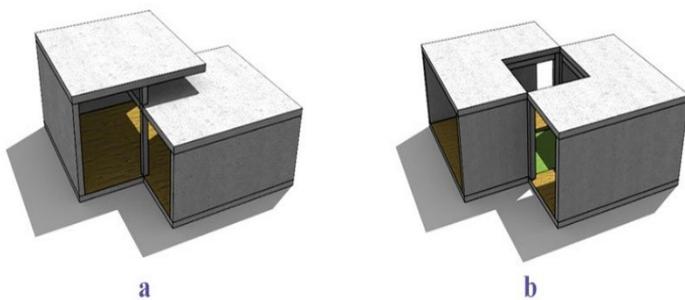


Figura 4: Perspectivas mostrando modelos 3D de diferentes interpretações de croquis.

xidade operacional de interação WISIWIT (*what I see is what I type*) o limitava a computadores especializados (estações de trabalho UNIX, equipados com dispositivos de entrada como mesas digitalizadoras, similares aos atuais *tablets* opacos), aos quais a população representativa de projetistas não tinha acesso. O sucesso na adoção dos sistemas operacionais com interfaces gráficas WIMP (*windows, icons, menus and pointer*) 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 muita da funcionalidade presente no CATIA, só com teclado e *mouse* (COHN, 2010). Desta maneira permitiu a sua adoção massiva, definindo um padrão de trabalho na indústria da arquitetura e engenharia de construção civil (*ibid.*).

O conhecimento técnico sobre métodos de cálculo e representação destes profissionais foi transferido para os *softwares* CAD com uma interação de aproximação WYSIWYG (*what you see is what you get*), que passaram a ser usados adaptando as técnicas típicas das pranchetas de desenho (BÜRDEK, 2005). Porém, 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 croquis (BUXTON, 1986). O que começou como um sucesso de adaptação às limitações de hardware acabou estagnando a evolução das metodologias digitais de projeto. (BHAVNANI; JOHN, 2001)

Os primeiros estudos sobre a usabilidade dos *softwares* CAD já demonstravam grandes problemas de desempenho dos projetistas (BHAVNANI; GARRETT; 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 importantes. Este fenômeno, conhecido como sobrecarga de funções⁴ (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.*).

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; DARK, 2013), como é o caso das tarefas de projeto. É o 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, é necessário interromper o raciocínio sobre elementos arquitetônicos e abstraí-lo, focando no protocolo ilustrado na Figura 5.

Figura 5: 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: Imagem cedida pelo 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 mudado o símbolo de representação de ponto nas opções do AutoCAD (do contrário, a representação do ponto por omissão, ou por *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 minutos atrás.

Interação das simulações de desempenho

Conhecimentos relacionados ao desempenho da edificação são aplicados nas tomadas de decisões, e externalizados 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).

Entre o suporte oferecido através de SketchUp, encontra-se a vinculação a sistemas de análises ambientais, como a extensão OpenStudio. Com esta extensão é possível usar o modelo 3D como informação de entrada para as análises do OpenStudio, apoiado no motor de cálculos ambientais EnergyPlus⁵. 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 no modelo 3D. 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 (CRAWLEY, HAND *et al.*, 2008). O projetista depende da sua capacidade de interpretação dos resultados para relacioná-los com o modelo e identificar nele os problemas de desempenho levantados pela análise.

As simulações de desempenho mudam temporariamente a aparência do modelo de maneiras diferentes, segundo o fenômeno a simular: o fluxo de ar é representado desenhando um campo de vetores; a simulação de iluminação envolve a discretização do modelo por amostragem e coloração de malhas poligonais, segundo a incidência da luz em cada ponto amostrado. (BUENO; TURKIENICZ, 2014)

Até 2014 era comum o uso de *softwares* de análise ambiental como Autodesk Ecotect e Autodesk Vasari. Ecotect e Vasari ofereciam funcionalidades, não só de análise, mas também de simulação de desempenho. Entre os fatores ambientais sobre os quais Ecotect oferecia estas funcionalidades encontravam-se: desempenho térmico, radiação solar, impacto visual, sombras e reflexões, e iluminação natural. Para todas estas simulações, Ecotect gerava uma malha poligonal discretizada, colorida em função dos valores de desempenho. A interação que o projetista tinha com esta simulação não era imediata. O modelo devia ser preparado, seja modelando diretamente no Ecotect ou importando um modelo elaborado em outros *softwares*, como SketchUp ou Rhinoceros (VANNINI, 2011). Esta preparação também sofria dos problemas de distanciamento

do foco da atividade, como foi previamente levantado, com o qual, o potencial de melhoria do desempenho do modelo não é suficientemente explorado. (GRABNER; FRICK, 2013)

Vannini (2011) estudou uma aplicação tecnológica de *softwares* que vincula automaticamente a simulação de desempenho de radiação solar do Ecotect com modelagem paramétrica e otimização por algoritmo genético (GA) em Grasshopper, um aplicativo de modelagem algorítmica para McNeel Rhinoceros. Por meio do domínio de habilidades de modelagem algorítmica e controle de parâmetros do GA, o projetista se apoia nas capacidades de processamento do computador para aumentar as suas habilidades projetuais. (RUTTEN, 2013)

Esta vinculação automática tem o potencial para encurtar a interação cíclica entre o projetista e o modelo simulado (GRABNER; FRICK, 2013), mas essa otimização 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. 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 GA (*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 do 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 automação. 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.

Outros conceitos de interação a considerar

Entrada 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*. 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.

Interação bimanual: modo de interação humano-computador na qual o usuário utiliza as duas mãos para controlar dispositivos de entrada.

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.

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.

O estado da arte do desenvolvimento de ferramentas para as etapas iniciais

Muitos dos trabalhos acadêmicos têm aportado avanços no campo através da implementação destas funcionalidades em *softwares* experimentais. Estudam-se treze, 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 interação permitida pelo sistema de entrada abordado, as funcionalidades de edição geométrica e as características de informação geométrica em relação com a interoperabilidade relevante para o ofício do arquiteto.

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.

Todos os *softwares* experimentais comparados se baseiam em aquisição de croquis e, com exceção de Kallio (2005) e Dorsey, Xu *et al.* (2007), todos trabalham com algum tipo de reconhecimento de formas.

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 croquis. 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*. A maioria define interação de desenho com a caneta, porém Kallio (2005) e Dorsey, Xu *et al.* (2007) dependem do teclado para controlar a posição 3D. Kang, Kim *et al.* (2013) suportam o desenho com os dedos, com gestos *multi-touch*. Lopes, Mendes *et al.* (2011) exploram uma interação bimanual multimodal, combinando caneta e *multi-touch*, e Pranovich (2004) adapta desenho vetorial 2D para o desenvolvimento de plantas arquitetônicas esquemáticas. Company, Contero *et al.* (2003) e Masry e Lipson (2005) implementam um complexo algo-

Referência	Área de aplicação	Entrada de desenho	Aproximação de desenho	Ação de reconstrução	Simulação de desempenho	Interoperabilidade
Company, Contero <i>et al.</i> , 2003	Engenharia	Caneta	Reconhecimento de perspectiva	Automática	--	Alta (DXF)
Pranovich, 2004	Arquitetura	Mouse e teclado	Desenho vetorial 2D	--	--	--
Juchmes, Leclerq e Azar, 2004	Arquitetura	Caneta	Desenho de croquis 2D	Automática	--	--
Kallio, 2005	Arquitetura e Design	Caneta e teclado	Desenho de croquis 3D	--	--	Média (VRML)
Masry e Lipson, 2005	Engenharia	Caneta	Reconhecimento de perspectiva	Automática	Estrutural	--
Oh, Stuerzlinger e Danahy, 2006	Arquitetura	Mouse e teclado	Extrusão de croquis 3D	WIMP	--	Média (3D)
Yu e Zhang, 2007	Arquitetura	Mouse	Extrusão de croquis 2D	WIMP	--	--
Dorsey, Xu <i>et al.</i> , 2007	Arquitetura	Caneta e teclado	Desenho de croquis 3D	--	--	Baixa (2D)
Elsen e Leclerq, 2008	Arquitetura	Caneta	Desenho de croquis 2D	--	--	--
Naya, Contero <i>et al.</i> , 2008	Engenharia	Caneta	Desenho de croquis 3D	Gestual	--	Alta (DXF)
Lopes, Mendes <i>et al.</i> , 2011	Artes Visuais	Caneta e multi-touch	Desenho de croquis 3D	Gestual	--	Alta (3D)
Elsen, Demaret <i>et al.</i> , 2012	Arquitetura	Caneta	Desenho de croquis 2D	Automática	--	--
Kang, Kim <i>et al.</i> , 2013	Engenharia	Multi-touch	Desenho de croquis 3D	Gestual	--	Média (macro)

Fonte: Imagem cedida pelo autor.

Tabela 2: Comparação das principais características dos softwares experimentais estudados.

ritmo para reconhecer formas 3D automaticamente desde croquis em perspectiva, enquanto Juchmes, Leclerq e Azar (2004) e Elsen, Demaret *et al.* (2012) abordam a reconstrução automática desde croquis em planta.

Kallio (2005) e Dorsey, Xu *et al.* (2007) não usam geometria sólida, mas croquis em 3D, evitando a implementação de rotinas de reconstrução. A construção de geometria sólida proposta por Oh, Stuerzlinger e Danahy (2006) e Yu e Zhang (2007) envolve a extrusão de polígonos reconhecidos, pela interação com recursos de interface WIMP (principalmente bo-

tões). Naya, Contero *et al.* (2008), Lopes, Mendes *et al.* (2011) e Kang, Kim *et al.* (2013) suportam gestos para o controle de reconstrução do croquis. Excetuando Masry e Lipson (2005), simulações preliminares de desempenho não são exploradas.

Company, Contero *et al.* (2003) e Naya, Contero *et al.* (2008) oferecem alta interoperabilidade usando o formato DXF, enquanto Lopes, Mendes *et al.* (2011) oferece alta interoperabilidade com outros formatos 3D. Kallio (2005) trabalha em formato VRML, limitando-se a linhas 3D. Oh, Stuerzlinger e Danahy (2006) permitem simulação de desempenho externas a partir de modelos exportados, enquanto Dorsey, Xu *et al.* (2007) permitem a geração de imagens em perspectiva que não podem ser utilizadas em etapas posteriores.

Cenários de utilização

A análise comparativa permitiu o desenvolvimento de cenários hipotéticos de uso de um *software* que oferecesse as funcionalidades estudadas. Bueno e Turkienicz (2014) apresentam num *storyboard* um cenário onde o projetista começa de zero, desenhando com a caneta *stylus* um croquis de formas geométricas relacionadas a um programa de necessidades. Após um breve tempo de inatividade, o *software* reconhece as formas como polígonos e gera uma nova camada de representação. O projetista entende os polígonos como contornos de edificação e gera os volumes a partir de gestos multimodais bimanuais (LOPES; MENDES *et al.*, 2011). Outros gestos *multi-touch* de navegação ajudam a orientar a vista e definir o âmbito das transformações geométricas: alterações do contorno em planta, e da altura em perspectiva.

Numa segunda etapa, o usuário desenha uma linha de corte, cuja simbologia é reconhecida pelo *softwa-*

re, que a cor de vermelho (cor dos símbolos por *default*). Um toque duplo ativa o modo de edição de corte, ortogonal com linhas em corte e em vista. O redesenho com a caneta é interpretado como alterações na geometria. Voltando à planta, o projetista desenha um símbolo de norte. Quando o símbolo é reconhecido, e coincide com a biblioteca de símbolos incorporada no *software*, este o interpreta como o sistema de coordenadas, dando a orientação geográfica necessária para simular o desempenho com respeito a fatores ambientais, como o sombreamento. (BUENO; TURKIENICZ, 2014)

Observando um problema de insolação excessiva numa fachada, numa terceira etapa, o projetista faz ajustes preliminares de desempenho. Com gesto bimanual, rotaciona as edificações em planta. Voltando à perspectiva, o modelo exhibe o novo sombreamento. O projeto avança numa quarta etapa de desenho de fachadas. Num espaço em branco, o usuário desenha um sólido, que logo mapeia sobre a superfície de uma das edificações com gestos bimanuais, e exporta para um formato de transferência, para dar início a seguinte fase de projeto em outro *software* (*ibid.*).

De maneira diferente ilustra-se outro cenário (*storyboard 2*), no qual se consideram outras funcionalidades.

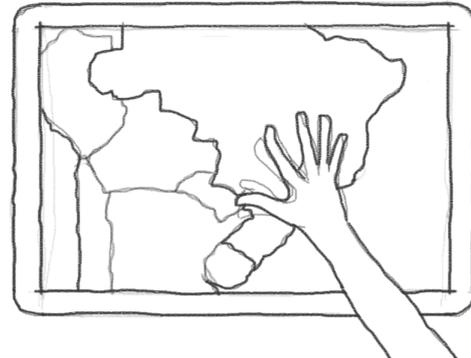
Etapa 1: Definição de contexto geográfico e urbano.

O projetista abre o 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 inter-

net. 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 6) 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 por um 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 croquis, 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 croquis).

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 po-



Fonte: Imagem cedida pelo autor.

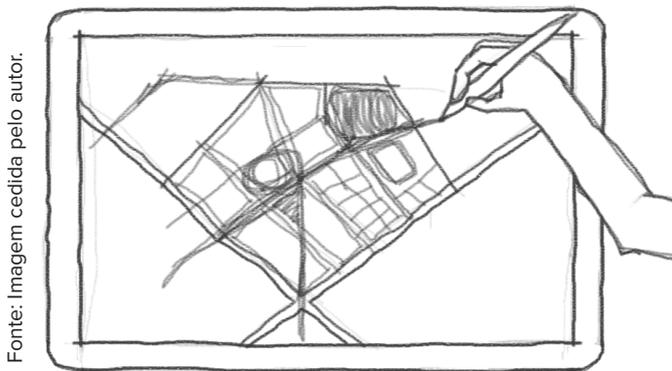
Figura 6: Storyboard 2: Navegação multi-touch pelo mapa terráqueo para definição de localização do projeto.

sicionada 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 quer incluir as construções existentes dentro do polígono ou desconsiderá-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 croquis, rabiscando vários traços de dimensões e pesos diversos. O *software* identifica, pela morfologia e densidade dos traços, que se trata de um croquis de pensamento (COMPANY, CONTERO *et al.*, 2006), no qual o projetista está usando o *feedback* do croquis para abordar problemas mal definidos. Por isto, o *software* mantém visível a representação do croquis, em vez da representação da geometria sendo interpretada (Figura 7).



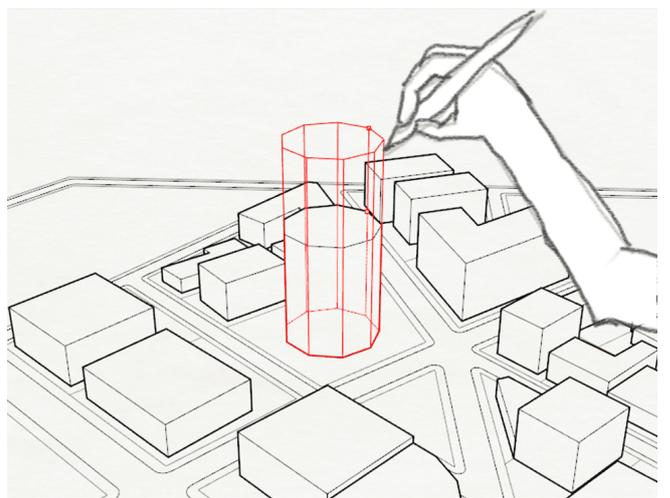
Fonte: Imagem cedida pelo autor.

Figura 7: Storyboard 2: Desenho de um croquis de pensamento sendo adquirido pelo software sem interrupções.

Pela escala do croquis, os traços são interpretados como elementos de desenho urbano: eixos de vias, perímetro limítrofe, vias, quadras, calçadas, parcelas e contornos de edificações. 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, o usuário edita a altura dos edifícios reconstruídos para satisfazer uma condição que 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 8), como em Oh, Stuerzlinger e Dahany (2006), e em Yu e Zhang (2007).



Fonte: Imagem cedida pelo autor.

Figura 8: Storyboard 2: Edição manual de uma extrusão, para modificar a altura de um edifício reconstruído.

Etapa 3: Simulação preliminar de desempenho ambiental.

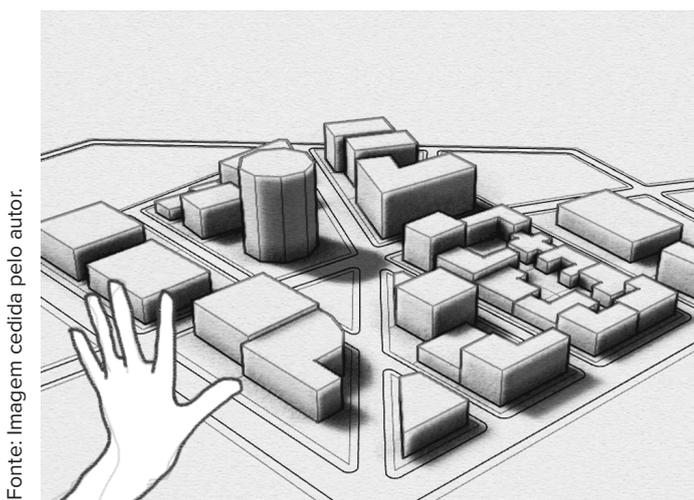
Para avaliar o impacto no conforto, o projetista se vale das funcionalidades de simulação de desempenho de alguns fatores ambientais. O primeiro a ser simulado é o de sombreamento (TURKIENICZ; GONÇALVES; GRAZZIOTIN, 2008), que é mostrado na tela após o usuário visualizar o modelo em perspectiva e afastar as mãos da tela por mais de quatro segundos (Figura 9a). 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 9b).

Este processo de modificação geométrica e a atualização das simulações de desempenho se repete 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*).

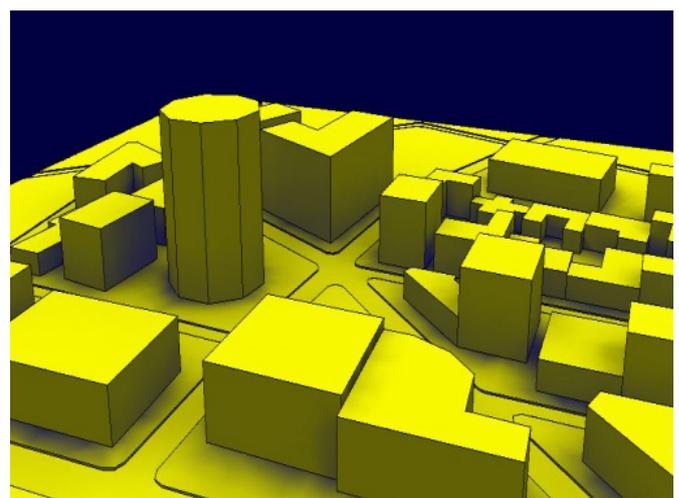
Síntese de funcionalidades sugeridas nos cenários de utilização

A Tabela 3, 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, como ele atua em função destes pensamentos, e qual a funcionalidade com a qual o *software* deveria responder.

Figura 9: Storyboard 2: Simulação de sombreamento (a) e de iluminância (b) sobre os edifícios e o espaço urbano sendo projetado.



(a)



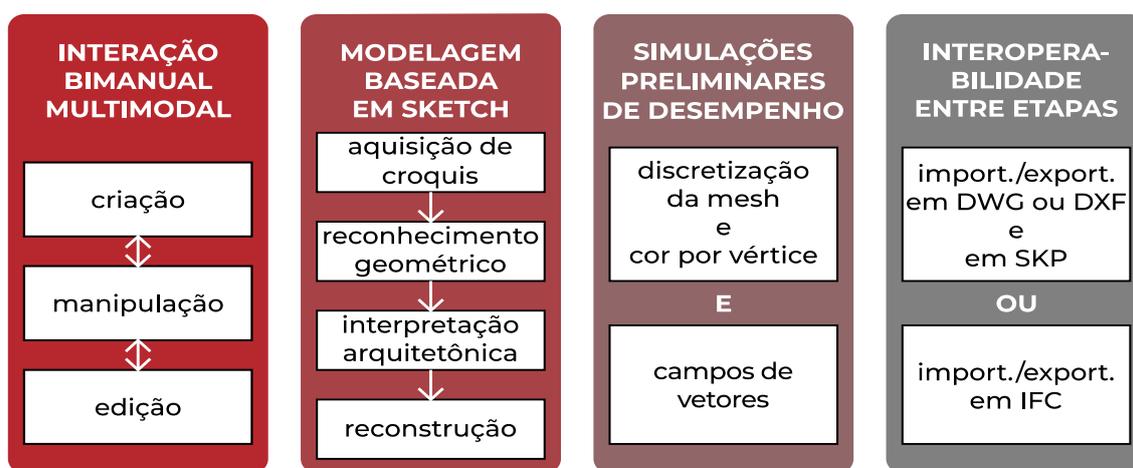
(b)

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 multi-touch de navegação; gesto de Extrusão.	Navegação: Orbital, Pan, Zoom; construção de sólidos por Extrusão.
Geração de alternativas; Modificação de relações entre formas em 2D e entre volumes de edificação; avaliação comparativa.	Gestos multi-touch de navegação; gestos multimodais de transformações.	Transições entre vistas em planta e em perspectiva; transformações: mover, rotacionar, escalar, esticar.
Atribuir 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 croquis; gestos multi-touch 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 multi-touch de navegação.	Amostragem da mesh; cálculo de Oclusão e Sombreamento; atribuição de cores de vértices, segundo valores de oclusão; navegação: Orbital, Pan, Zoom.
Modificações globais para solucionar problemas de sombreamento entre edificações.	Gestos multi-touch 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, Pan, Zoom.
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 croquis.	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 multi-touch de navegação e seleção; gestos multimodais de transformações avançadas.	Reparametrização de superfícies; duplicação em matriz sobre superfícies.

Tabela 3: Interações e processamento segundo processos cognitivos, sugeridos nos cenários de uso.

Conclusões

Toda a funcionalidade descrita deveria ser integrada num só *software* para o apoio apropriado às etapas iniciais de projeto arquitetônico. Estas funcionalidades têm papéis ou atributos específicos, necessários para estas etapas e são apresentados sinteticamente na Figura 10.



Fonte: Imagem cedida pelo autor.

Figura 10: Princípios de funcionalidade necessários em softwares de apoio nas etapas iniciais de projeto de arquitetura.

Princípios de funcionalidade para apoiar as etapas iniciais de projeto arquitetônico

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*; **(III) Edição:** modificar o modelo, por meio da combinação simultânea de caneta e dedos que define gestos bimanuais para funções de edição e trans-

formações geométricas. Na necessidade de escolha de implementação de funcionalidades cujos gestos sejam conflitivos, provindo uns de *softwares* preexistentes e outros de ações manuais, típicas de projeto arquitetônico, considerar os últimos.

Modelagem baseada em sketches

Criar e editar desenhos de projeto baseados em traços à mão livre no espaço de trabalho que geram modelos em 3D. A criação deve ser baseada em quatro módulos de: **(I) Aquisição de croquis:** obter, da entrada do usuário que utiliza uma caneta *stylus*, dados vetoriais do traço na tela como sequências de linhas. Registrar as variações de pressão imprimida com a

caneta sobre a tela, para futuro uso; **(II) Reconhecimento geométrico:** analisar os croquis adquiridos para registrar formas primitivas e outros dados básicos. Essencialmente, distinguir entre traços que representam geometria da edificação e traços que representam gráficos complementares (símbolos arquitetônicos, gestos comunicativos e anotações). Focar no reconhecimento de geometria em 2D, já que é a mais utilizada entre arquitetos. Usar a variação de pressão registrada na aquisição para ajudar na discriminação de formas e definir tipos de linhas. Incluir a intervenção do usuário neste processo para aproveitar sua inteligência visual para reconhecer formas emergentes; **(III) Interpretação arquitetônica:** computar a geometria reconhecida como o elemento arquitetônico em 3D 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, gerar o modelo 3D através de processos automáticos, basicamente a partir de geometria arquitetônica em 2D, seguindo as regras da linguagem gráfica do arquiteto. Complementar com reconstrução gestual para lidar com ambiguidades

Simulações preliminares de desempenho

Modificar temporariamente a aparência do modelo para representar o desempenho preliminar da edificação sendo projetada. Exibir uma representação não fotorealística do projeto e substituir esta visualização quando valores de desempenho estejam disponíveis. Em momentos de inatividade, ou por requisição do usuário, analisar o desempenho e exibir os resultados diretamente no modelo, amostrando-o com uma malha poligonal, na qual os vértices sejam

coloridos (por exemplo, radiação solar) ou movidos (por exemplo, deformações estruturais) simulando as consequências físicas para cada vértice. Simulações preliminares de desempenho devem ser aproximações simplificadas para garantir uma rápida execução e para responder adequadamente aos problemas mal definidos das etapas iniciais de projeto.

Interoperabilidade entre etapas de projeto

Os formatos de arquivos devem atender padrões da indústria para oferecer interoperabilidade com os *softwares* mais usados nas etapas avançadas de projeto, 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. Os padrões *de facto* da indústria ainda são 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.

É notória a importância da aplicação que um *software* como o imaginado nos cenários de uso tem na evolução das técnicas de projeto de arquitetura e urbanismo. Sem o suporte integral às atividades mais importantes das etapas iniciais de projeto, o arquiteto continuará preferindo ferramentas tradicionais como o lápis e papel em vez das tecnologias digitais. A falta de suporte destas funcionalidades desaproveita estas tecnologias, não atendendo às demandas do ofício, por projetos cada vez mais eficientes com prazos cada vez menores.

Notas de Fim:

1. Artigo desenvolvido a partir da dissertação de mestrado do autor no Programa de Pós-Graduação em Design da Universidade Federal do Rio Grande do Sul (PGDESIGN/UFRGS), sob orientação do professor Dr. Benamy Turkienicz. Parte do trabalho foi publicado em inglês no *International Journal of Architectural Computing* (BUENO; TURKIENICZ, 2014). Gostaria de agradecer ao Prof. Benamy pela valiosa orientação. Este trabalho foi realizado com apoio parcial da CAPES - Brasil, e com apoio parcial da FAURGS.

2. 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; YAN, 1993) (OXMAN, 2002).

3. 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).

4. Do termo em inglês: *function overload*, trazido ao discurso do projeto 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).

5. 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.

Referências bibliográficas

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. BUENO, Ernesto. *Uso de tecnologias digitais nas etapas iniciais de projeto arquitetônico*. 2014. Dissertação (Mestrado em Design) – Faculdade de Arquitetura – Escola de Engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre. Disponível em: <<http://hdl.handle.net/10183/130093>>.

BUENO, E.; TURKIENICZ, B. Supporting Tools for Early Stages of Architectural Design. *International Journal of Architectural Computing*, v. 12, n. 4, p. 495-512, 2014. DOI: 10.1260/1478-0771.12.4.495 BÜRDEK, B. *Design: History, Theory and Practice of Product Design*. Berlin: Birkhäuser Publishers for Architecture, 2005.

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. _____. Multi-touch Systems that I Have Known and Loved. *Bill Buxton Home Page*, 19 mar 2013. Disponível em: <www.billbuxton.com/multitouchOverview.html>. Acesso em: 2013-11-30.

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: 2013-11-20. Section 10.

COHN, D. Evolution of Computer-Aided Design: How we got to where we are, and where are we headed. *Desktop Engineering*, 2010-12-01. Disponível em: <www.desktopeng.com/articles/aaazer.htm>. Acesso em: 2014-02-07.

COMPANY, P.; CONTERO, M. et al., Aplicación docente de un programa de modelado 3D mediante bocetos axonométricos. In: *XIII ADM – XV INGEGRAF International Conference on Tools and Methods Evolution in Engineering Design*. Napoli: ADM. 2003.

- _____. A study of usability of sketching tools aimed at supporting prescriptive sketches. In: *Proceedings of the Third Eurographics conference on Sketch-Based Interfaces and Modeling*. Aire-la-Ville: Eurographics Association. 2006. p. 139-146. DOI: 10.2312/SBM/SBM06/139-146
- CRAWLEY, D.; HAND, et al. Contrasting the capabilities of building energy performance simulation programs. *Building and Environment*, v. 43, n. 4, p. 661-673, 2008. DOI: 10.1016/j.buildenv.2006.10.027
- DENZER, A.; GARDZELEWSKI, J. Drawing and Modeling: Analog Tools in the Age of BIM. In: *Proceedings of the 2011 Architectural Engineering Conference*. Oakland (CA): American Society of Civil Engineers. 2011. p. 44-53.
- DORSEY, J.; XU, et al. The mental canvas: A tool for conceptual architectural design and analysis. In: *15th Pacific Conference on Computer Graphics and Applications, PG'07*. [S.l.]: IEEE. 2007. p. 201-210.
- ELSEN, C.; DEMARET, J. 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. DOI: 10.1017/S0890060412000157
- ELSEN, C.; LECLERCQ, P. "SketSha" - The Sketch Power to Support Collaborative Design. In: LUO, Y. *Co-operative Design, Visualization, and Engineering*. Berlin: Springer, 2008. p. 20-27. Lecture Notes in Computer Science, v. 5220. DOI: 10.1007/978-3-540-88011-0_3
- 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*, v. 83, n. 2, p. 142-143, 2013. DOI: 10.1002/ad.1572
- HOFFMAN, D. *Visual Intelligence: How we create what we see*. [S.l.]: W.W. Norton & Company, 1998.
- JUCHMES, R.; LECLERCQ, P.; AZAR, S. A freehand-sketch environment for architectural design supported by a multi-agent system. *Computers & Graphics*, v. 29, n. 6, p. 905-915, 2005.
- KALLIO, K. 3D6B Editor: Projective 3D Sketching with Line-Based Rendering. In: *Proceedings of the 2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling*. Dublin: Eurographics Association. 2005. p. 73-79.
- KANG, Y.; KIM, H. et al. Feature-based 3D CAD Modeling on Smart Device Using Multi-touch Gesture. *International Journal of CAD/CAM*, v. 13, n. 2, p. 49-62, 2013.
- LAWSON, B. *How Designers Think: The Design Process Demystified*. Oxford (UK): Elsevier Architectural Press, 2005.
- LOPES, P.; MENDES, D. et al. Combining bimanual manipulation and pen-based input for 3D modelling. In: *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*. Annecy: Eurographics Association. 2011. p. 15-22. DOI: 10.2312/SBM/SBM11/015-022
- MASRY, M.; LIPSON, H. A Sketch-Based Interface for Iterative Design and Analysis of 3D Objects. In: *Proceedings of the 2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling*. Dublin: Eurographics Association. 2005. p. 109-118. DOI: 10.2312/SBM/SBM05/109-118
- MEHTA, N. *A Flexible Machine Interface*. University of Toronto. Toronto, 1982.
- MOREIRA, D.; KOWALTOWSKI, D. Discussão sobre a importância do programa de necessidades no processo de projeto em arquitetura. *Ambiente Construído*, v. 9, n. 2, p. 31-45, 2009.
- MURUGAPPAN, S.; RAMANI, K. FEAsy: A Sketch-Based Interface Integrating Structural Analysis in Early Design. In: *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. San Diego: ASME. 2009. p. 743-752.
- NAYA, F.; CONTERO, M. et al. Sketch-Based Interfaces for Parametric Modelling. In: PAVLIDIS, I. *Human Computer Interaction*. [S.l.]: InTech, 2008. p. 43-56. DOI: 10.5772/6310
- OH, J.; STUERZLINGER, W.; DANAHY, J. Sesame: Towards better 3D conceptual design systems. In: *Proceedings of the 6th Conference on Designing Interactive Systems*. New York: The ACM Press. 2006. p. 80-89.

OXMANN, R. Design by re-representation: a model of visual reasoning in design. *Design Studies*, v. 18, n. 4, p. 329-347, 1997. DOI: 10.1016/S0142-694X(97)00005-7

_____. The thinking eye: visual re-cognition in design emergence. *Design Studies*, v. 23, n. 2, 2002. DOI: 10.1016/S0142-694X(01)00026-6

OXMANN, R. Performative Design: A Performance-Based Model of Digital Architectural Design. *Environment and Planning B: Planning & Design*, v. 36, n. 6, p. 1026-1037, 2009. DOI: 10.1068/b34149

PRANOVICH, S. *Structural Sketcher: A Tool for Supporting Architects in Early Stages*. Technische Universiteit Eindhoven. Eindhoven, p. 150. 2004.

RUTTEN, D. Galapagos: On the logic and limitations of generic solvers. *Architectural Design*, v. 83, n. 2, p. 132-135, 2013. DOI: 10.1002/ad.1568

STILL, J.; DARK, V. Cognitively describing and designing affordances. *Design Studies*, v. 34, n. 3, p. 285-301, 2013. DOI: 10.1016/j.destud.2012.11.005

SUWA, M.; TVERSKY, B. What architects and students perceive in their sketches: A protocol analysis. *Design Studies*, v. 18, n. 4, p. 385-403, 1997. DOI: 10.1016/S0142-694X(97)00008-2

TURKIENICZ, B.; GONÇALVES, B.; GRAZZIOTTIN, P. CityZoom: A Visualization Tool for the Assessment of Planning Regulations. *International Journal of Architectural Computing*, v. 6, n. 1, p. 79-95, 2008. DOI: 10.1260/147807708784640144

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*. 2011. Dissertação (Mestrado em Arquitetura) – Universidade Federal do Rio Grande do Sul. Porto Alegre. Disponível em: <<http://hdl.handle.net/10183/33454>>.

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

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

WOO, J.; CLAYTON *et al.*, Dynamic Knowledge Map: reusing experts' tacit knowledge in the AEC industry. *Automation in Construction*, v. 13, n. 2, Mar 2004. 203-207. DOI: 10.1016/j.autcon.2003.09.003

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