

Blickwinkelabhängige Interaktion auf mehreren Anzeigegeräten mithilfe des Perspective Cursors

Proseminarbeitrag im Rahmen des Proseminars „Mensch-Computer-Interaktion“ im Sommersemester 2007 an der Universität Ulm

Benjamin Erb

Universität Ulm
Fakultät für Informatik
Albert-Einstein-Allee 11
D-89069 Ulm
benjamin.erb@uni-ulm.de

Zusammenfassung Da Arbeitsumgebungen mit mehreren Bildschirmen und verschiedenen, zum Teil auch mobilen Geräten immer verbreiteter werden, steigt der Wunsch, auch nach Vernetzung der einzelnen Geräte weiterhin intuitiv und einfach auf der entstanden gemeinsamen Bildfläche zu interagieren. Aufgrund von unterschiedlichen Abmessungen, Auflösungen, Anordnungen und Positionen im Raum entsteht jedoch eine große Heterogenität der einzelnen Bildflächen, so dass klassische Stitching-Techniken dem Benutzer kaum eine natürliche Navigation ermöglichen, da hier immer von einer ebenen Bildflächenanordnung ausgegangen wird. Blickwinkelbasierte Interaktionstechniken hingegen nutzen Perspektive als Hilfsmittel, um Räumlichkeit miteinzubeziehen. Bisherige Interaktionsgeräte hierzu besitzen aber Nachteile insbesondere hinsichtlich der Genauigkeit und Ergonomie. Der *Perspective Cursor* versucht die Vorteile eines relativ positionierenden Eingabegerätes mit den Vorteilen einer blickwinkelbasierten Interaktionstechnik zu verknüpfen, um in heterogenen Bildschirmumgebungen intuitive und natürliche Interaktionen zu ermöglichen.

1 Einleitung und Problemstellung

Nicht nur Größe, Auflösung und Bildqualität aktueller Monitore nimmt immer weiter zu, auch Arbeitsumgebungen mit mehreren Bildschirmen werden immer mehr eingesetzt. Dabei ist deren Anwendung nicht mehr nur auf spezielle Überwachungs-, Kontroll- oder etwa CAD-Arbeitsplätze beschränkt. Zugleich nimmt auch die Anzahl an mobilen Geräten mit Bildschirmen wie Laptops, Tablet-PCs, PDAs oder Handys zu, und die Grenzen zwischen ihnen verwischen durch Ubiquitous Computing immer mehr. Außerdem drängen neue Interaktionstechnologien wie Wand- oder Tisch-PCs [2] auf den Markt. In verschiedenen Kontexten ist die gemeinsame und einheitliche Nutzung der verschiedenen Geräte sinnvoll und wünschenswert. Zwar lassen sich all diese Geräte physikalisch auf verschiedene Weise miteinander verbinden [4], jedoch stellen sich Interaktionen vor allem zwischen den Bildflächen dieser verknüpften Einheiten als eher

kompliziert heraus. Grund dafür ist die Schwierigkeit, eine kohärente Abbildung zwischen der virtuellen Arbeitsfläche und den physikalischen Bildflächen herzustellen. Bisherige Techniken zur Interaktion auf mehreren Anzeigeflächen wie das *Stitching* versuchen die verschiedenen Anzeigen virtuell auf einer zweidimensionalen Fläche anzuordnen und „zusammenzunähen“. Dies funktioniert aber nur, solange die Anzeigegeräte tatsächlich plan zueinander stehen. Denn die Nichtbeachtung räumlicher Gegebenheiten und Aufteilungen führt zu einer unnatürlichen, nicht intuitiven und erschwerten Bedienung, insbesondere beim Übergang des Zeigers zwischen zwei Bildflächen, also an den Nähten. Auch werden durch Entfernung oder Auflösung entstandene Größenrelationen nicht berücksichtigt.

Blickwinkelabhängige Techniken nutzen hier die räumlichen Verhältnisse zur Abbildung der Bildflächen, indem sie von der Position des Benutzers aus die gegebenen räumlichen Verhältnisse auf eine ebene Darstellung abbilden - sich also die Perspektive zu nutze machen. Ein typisches blickwinkelabhängiges Werkzeug ist der Laser-Pointer. Da für Menschen das Zeigen ein intuitiver Vorgang ist, fällt es nicht schwer, mit einem Pointer an bestimmte Stellen im Raum zu deuten und dort zu navigieren. Außerdem geht der Pointer aufgrund der Perspektive auf die räumliche Tiefe ein. Auf nahen Flächen kann exakter navigiert werden als auf weiter entfernten. Allerdings ist diese Methode aufgrund fehlender Genauigkeit und einer hohen motorischen Beanspruchung des Benutzers nicht optimal.

Als Kompromiss zwischen diesen beiden Techniken soll der *Perspective Cursor* eine blickwinkelabhängige Interaktionstechnik sein, die die Vorteile der Perspektive mit den Vorteilen eines relativ positionierenden Eingabegerätes verbindet und ein nahtloses und intuitives Bewegen eines Zeigers auf räumlich verteilten Bildflächen ermöglicht. Der *Perspective Cursor* wird im Folgenden basierend auf einer Veröffentlichung [3] von Miguel A. Nacenta, Samer Sallam, Bernard Champoux, Sriram Subramanian und Carl Gutwin von der Universität von Saskatchewan vorgestellt.

2 Techniken zur Abbildung virtueller Arbeitsflächen auf mehrere physikalische Anzeigen

Das grundlegende Problem bei der Abbildung einer virtuellen, zweidimensionalen Bildfläche auf mehrere im Raum stehende physikalische ist es, eine Repräsentation der Flächen zu finden, mit der der Benutzer intuitiv zwischen den Teilflächen interagieren kann.

2.1 Stitching

Stitching ist eine Technik, die in vielen aktuellen Betriebssystemen zum Einsatz kommt, wenn mehrere Anzeigegeräte genutzt werden. Die virtuelle Bildfläche wird aufgeteilt in Teilbildflächen, die auf den einzelnen Monitoren dargestellt werden. Hierfür muss der Benutzer anfangs die *Stitching*-Funktionalität so konfigurieren, dass die Position der einzelnen Teilbildflächen möglichst der räumlichen Position der Anzeigegeräte entspricht. Durch *Stitching* werden dann die Teilbildflächen zu einer Gesamtfläche zusammengekachelt. Wird der Zeiger über den Rand einer Teilbildfläche bewegt, so wird er

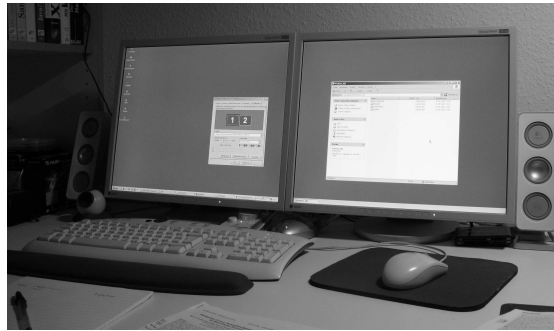


Abbildung 1. Eine typische Dual-Screen-Anordnung. Auf dem linken Monitor ist der Stitching-Konfigurationsdialog von Windows XP zu sehen.

nathlos auf die angrenzende Teilbildfläche gesetzt. Das heißt, dass der Zeiger dann von dem Rand einer Anzeige zu dem angrenzenden Rand der nächsten springt. Wird ein absoluter Rand erreicht, also ein Rand, an den keine weitere Anzeige angrenzt, so lässt sich der Zeiger nicht weiter in diese Richtung bewegen.

Stitching funktioniert gut, solange die Anzeigen benachbart auf einer gemeinsamen Ebene angeordnet sind (siehe Abb. 1). Dadurch, dass der Benutzer die Positionen einmal einstellen muss, muss allerdings nach jeder Änderung der räumlichen Anordnung der Anzeigegeräte dieser Vorgang erneut durchgeführt werden. Insbesondere im Zusammenspiel mit mobilen Geräten ist dies unbefriedigend. Da bisherige *Stitching*-Implementierungen nicht unterschiedliche Auflösungen der einzelnen Geräte oder die Breite der Bildschirmrahmen beachten, kann es zudem zu Sprüngen des Zeigers kommen, was den Benutzer verwirren und beeinträchtigen kann. Die Nichtbeachtung der räumlichen Gegebenheiten stellt sich als größtes Problem im Bezug auf die Nutzung von *Stitching* heraus. Dadurch, dass die Repräsentationen aller Bildflächen im Zweidimensionalen angeordnet werden müssen, kommt es in vielen Situationen zu Konflikten zwischen virtueller Repräsentation und echter Anordnung. So lassen sich weder überlappende, gedrehte oder weiter entfernte Anzeigen sauber virtuell anordnen, noch werden leere Zwischenräume im Sichtfeld des Benutzers beachtet.

2.2 Blickwinkelabhängige Techniken

Blickwinkelabhängige Interaktionstechniken für Umgebungen mit mehreren Bildflächen benutzen Perspektive als Hilfsmittel für eine bessere Abbildung von virtueller Repräsentation und echten Bildflächen. Hierbei werden Daten über die Position, Ausrichtung und Entfernung der einzelnen Anzeigegeräte sowie der Position des Benutzers gesammelt und miteinander verrechnet. Daraus kann ein virtuelles räumliches Modell errechnet werden, das anders als beim zweidimensionalen *Stitching* auch räumliche Anordnungen miteinbezieht. Bewegt der Benutzer nun sein Zeigegerät, so bewegt sich der Zeiger im virtuellen räumlichen Modell wie auch im echten, sichtbaren Raum gleichmäßig. Die perspektivische Herangehensweise hat neben der intuitiven

Handhabung noch einen weiteren Vorteil: Je näher eine Fläche ist, desto genauer lässt es sich auf ihr bei gleichbleibendem Aufwand navigieren. Da man mit einem blickwinkelabhängigen Eingabegerät nur die Anzeigegeräte steuern kann, auf die man auch zeigen kann, spricht man auch von *What you see is what you can control*.

Virtual Beam Das Eingabegerät *Virtual Beam* [3] orientiert sich eng an dem Modell eines Laser-Pointers. Anstatt eines Laserpunktes wird jedoch ein Zeiger dargestellt. Anhand der Ausrichtung des Pointers und den bereitstehenden Positionsdaten wird geprüft, ob sich auf der anvisierten Position ein Anzeigegerät befindet. Wenn ja, so wird auf diesem Anzeigegerät der Zeiger an der anvisierten Stelle dargestellt. Liegen mehrere Anzeigegeräte auf der Geraden, so wird gemäß der Perspektive das vorderste sichtbare ausgewählt. Da der Pointer nichts emittiert, wird nur dann ein Zeiger dargestellt, wenn er auf ein Anzeigegerät gerichtet ist. Für mausähnliche Interaktionen besitzt der Pointer mehrere Buttons. Allerdings erbt dieses Eingabegerät auch die Schwächen eines Laser-Pointers. Intensive Interaktionen über einen längeren Zeitraum führen zu einer hohen motorischen Beanspruchung des Benutzers. Außerdem nimmt insbesondere während Operationen wie dem Betätigen eines Buttons die Genauigkeit des Pointers ab.

Perspective Cursor Der von Miguel A. Nacenta, Samer Sallam, Bernard Champoux, Sriram Subramanian und Carl Gutwin von der Universität von Saskatchewan entwickelte *Perspective Cursor* versucht einen Kompromiss zwischen den beiden zuvor genannten Techniken zu bilden. Die Pointer-Eingabe wurde durch die Verwendung eines relativ positionierenden Eingabegerätes modifiziert. Außerdem wurde versucht, die Schwächen bisheriger blickwinkelabhängiger Techniken zu überwinden. Wie bei *Virtual Beam* werden ebenfalls die Positionen der Anzeigegeräte sowie des Benutzers geprüft. Anstatt jedoch mit einem Zeigeinstrument den Zeiger zu führen, wird eine imaginäre Linie zwischen der Position des Benutzers und der Position des Zeigers gebildet. Bewegt dieser sein relativ positionierendes Eingabegerät, zum Beispiel eine Maus, vertikal, so ändert sich der äquatoriale Winkel der Linie. Eine horizontale Bewegung der Maus bewirkt eine longitudinale Winkeländerung. Somit kann der Benutzer mit Mausbewegungen die Position des Zeigers im Raum gleichmäßig ändern. Trifft die imaginäre Linie auf ein Anzeigegerät, so wird der Zeiger dargestellt (*siehe Abb. 2*). Um aber auch bei übergehenden Bewegungen zwischen Anzeigegeräten dem Benutzer eine Rückmeldung über die aktuelle Position des Zeigers zu geben, wurde eine als Halo [1] bezeichnete Technik angepasst und verwendet. „Halo“ steht für Lichthof und wird in diesem Kontext als sichtbarer Ring um den Zeiger benutzt, wenn dieser selbst nicht mehr sichtbar ist. Verlässt der Zeiger den darstellbaren Bereich, so erscheinen auf den umliegenden Bildflächen Ausschnitte eines Ringes. Je weiter der Zeiger von einer Anzeigefläche entfernt ist, desto mehr wird der Ringausschnitt zu einer geraden Linie, je näher er ihr wieder kommt, desto mehr krümmt sich der Ringausschnitt wieder. Eine weitere Verbesserung betrifft die Darstellung des Zeigers. Aus den Positionsdaten wird eine solche Darstellung des Zeigers errechnet, dass dieser für den Benutzer eine unabhängig von der Position gleichmäßig wirkt. Das heißt, dass der Zeiger auf unterschiedlichen Bildflächen skaliert und zum Teil auch verzerrt wird, aber in der Retina des Benutzers eine feste Größe und einheitliche Form besitzt. Dies ist insbesondere bei

Kombinationen eines Tabletops und eines Beamers wichtig, also wenn die Bildflächen einen großen Winkel einschließen.

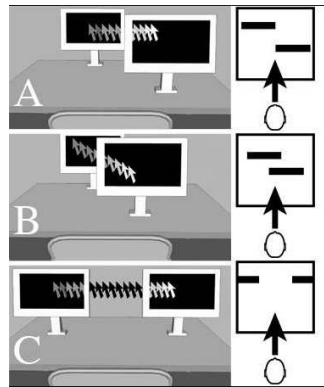


Abbildung 2. Beispiele für den *Perspective Cursor* [3]. Man beachte die Bewegungen und die konstante Größe des Zeigers in den unterschiedlichen räumlichen Situationen. In Illustration C würde bei einer Zeigerposition zwischen den beiden Monitoren auf den Monitoren Halo-Ringe erscheinen, um dem Benutzer bei der Navigation ohne sichtbaren Zeiger zu helfen.

3 Vergleich und Fazit

Nacenta und sein Team führten eine kleine Studie durch, um die drei genannten Eingabetechniken miteinander zu vergleichen. Die Testperson befand sich in einer heterogenen Anzeigenumgebung, bestehend aus einer Tabletop-Anzeige, einem an die Wand projizierten Beamer-Bild, einem Tablet-PC und einem gewöhnlichen Flachbildschirm. Dabei befanden sich Flachbildschirm und Tablet-PC auf der Tabletop-Anzeige, so dass es zu Überlappungen kam, außerdem konnte der Tablet-PC bewegt werden. Nun sollte die Testperson verschiedene Interaktionen durchführen, wie das Betätigen zweier auf der Fläche verteilter Symbole. Dabei wurden Punktepaare mit verschiedenen Schwierigkeitsstufen gewählt. So mussten zum Beispiel leere Flächen zwischen Anzeigegeräten überquert werden, oder es befanden sich auf der Bahn des Zeigers verschiedene überlappende Bildflächen. Dabei wurden benötigte Zeit, die Erfolgsrate sowie die Benutzerbelastung ausgewertet.

Als erstes Ergebnis konnte bestätigt werden, dass *Stitching*-Techniken bei einfachen Konstellationen der Anzeigegeräte befriedigende Ergebnisse hervorbrachte, allerdings bei komplexeren räumlichen Abläufen den Benutzer ausbremste. Die Testpersonen gaben im Nachhinein an, sich teilweise vor Interaktionen erst Gedanken gemacht zu haben, wie die Zeigerbewegungen umzusetzen zu seien. Andere berichteten davon, häufig nach einem Übergang zwischen zwei Bildflächen über das anvisierte Ziel herausgeschossen zu sein, da *Stitching* die nicht darstellbaren Zwischenflächen nicht beachtet.

Weiterhin unterstrich die Studie die Überlegenheit blickwinkelabhängiger Techniken. *Virtual Beam* arbeitete effizienter als *Stitching*, wenn es um komplexere räumliche Bewegungen ging. Insbesondere bei Testläufen mit vielen Überlappungen erreichte *Virtual Beam* gute Bewertungen. Allerdings besaß auch keine andere Technik eine solch hohe Rate an fehlerhaften Klicks, bei denen es den Testpersonen nicht gelang, das angezeigte Symbol exakt zu treffen. Auch gaben viele Testpersonen an, dass die Tests mit *Virtual Beam* sehr strapaziös waren.

Der Großteil der Testpersonen beurteilte den *Perspective Cursor* als angenehmste und einfachste Interaktionsmethode und auch bei der statistischen Auswertung konnte der *Perspective Cursor* einen Gesamtsieg erzielen, trotz vereinzelter Teilsiege anderer Techniken in speziellen Situationen.

Abschließend lässt sich feststellen, dass die Technik des *Perspective Cursors* eine fortschrittliche und weiter verfolgenswerte Idee zur Steuerung eines übergreifenden Zeigers in heterogenen Umgebungen zu sein scheint. Durch die Einarbeitung der Perspektivabhängigkeit ermöglicht sie den Anwendern schnelle, genaue, aber auch intuitive und natürliche Zeigerbewegungen trotz räumlicher Hindernisse.

Zwar findet kein Eye-Tracking des Benutzers statt, aber es ist immerhin notwendig, in Echtzeit die Kopfposition des Benutzers sowie Positionen und Anordnungen der einzelnen Anzeigegeräte zu bestimmen. Technologien, die diese Funktionalität bieten, sind bisher sehr teuer und verhindern noch eine kostengünstige Implementierung eines *Perspective Cursors*. Allerdings ist davon auszugehen, dass insbesondere Forschungen im Bereich Computer Vision und Tracking in naher Zukunft akzeptable Lösungen hierfür bieten könnten. Desweiteren rechtfertigen die kritischen Anwendungen bei klassischen, hochintegrierten Mehrfachbildschirmumgebungen wie technischen Kontrollräumen oder militärische Kommandozentren schon heute den Preis zusätzlicher Ausrüstung für eine intuitivere und effizientere Steuerung.

Das Team um Miguel A. Nacenta will sich weiterhin mit dem *Perspective Cursor* beschäftigen und sich dabei auf eine mögliche Anwendung für mehreren Benutzer und Arbeitsgruppen konzentrieren, aber auch den *Perspective Cursor* unter Aspekten wie Datenschutz und alternativer Implementierungsmöglichkeiten neu betrachten.

Literatur

1. Baudisch, P.; Rosenholtz, R.: Halo: a Technique for Visualizing Off-Screen Locations. CHI '03: Proceedings of the SIGCHI conference on Human Factors in computing systems (2003) 481-488
2. Microsoft Corporation: Microsoft Launches New Product Category: Surface Computing Comes to Life in Restaurants, Hotels, Retail Locations and Casino Resorts. <http://www.microsoft.com/presspass/press/2007/may07/05-29MSSurfacePR.mspx> (03.06.07)
3. Nacenta, M. A.; Sallam, S.; Champoux, B.; Subramanian, S.; Gutwin, C.: Perspective cursor: perspective-based interaction for multi-display environments. CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems (2006) 289-298
4. Rekimoto, J.: Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. ACM Symposium on User Interface Software and Technology (1997) 31-39