

FishEye-Views für den mobilen Zugriff auf Prozeßbilder

FishEye Views for the Mobile Access to Process Visualizations

Uwe Rauschenbach

*Siemens AG, Corporate Technology, CT IC 2
Otto-Hahn-Ring 6, 81730 München
Tel: +49 89 636 49751, Fax: +49 89 636 51115
Email: uwe.rauschenbach@mchp.siemens.de*

Kurzfassung: Bei der Darstellung von hoch aufgelösten Prozeßvisualisierungen auf den kleinen Displays mobiler Computer kann nur ein Ausschnitt dargestellt werden, so daß der Überblick über die Gesamtdarstellung verlorengeht. Dieser Beitrag beschreibt den Rechteckigen FishEye-View als Lösung für dieses Problem. Hierbei wird der momentan für den Benutzer wichtigste Bildausschnitt (der Fokus) detailliert dargestellt, und die umgebenden Bildbereiche (der Kontext) werden gestaucht, was zu einer Platzeinsparung führt. Auf Basis des Nutzerinteresses verändert das System dynamisch die Fokusposition. Die Eigenschaften der Darstellungstechnik ermöglichen eine schnelle Implementierung auch auf mobilen Geräten mit eingeschränkter Rechenleistung. Eine Realisierung in einem Thin-Client-System wird kurz vorgestellt.

Abstract: When displaying high-resolution process visualizations on mobile computers with limited screen space, only a clipping can be displayed, thus losing the overview over the whole visualization. This paper presents the Rectangular FishEye View as a solution for this problem. The from a user's point of view most important visualization part (the focus) is rendered at high detail, and the surrounding context is downscaled to save display space. Based on the user's interest, the system can adapt the focus position automatically. The properties of the technique allow for the fast implementation even on mobile devices with limited computing power. A realization in the context of a thin client system will be described.

1 Motivation und Problemstellung

Durch die Verfügbarkeit robuster, mobiler Tablet-Computer (z.B. MOBIC /1/) und drahtloser Netze (WLANs) rückt die technische Realisierbarkeit des mobilen, drahtlosen Zugriffs auf Leitwartendaten von beliebigen Punkten innerhalb einer Fabrikhalle in greifbare Nähe. Damit können relevante Informationen von den Bedienern, Einrichtern oder Servicekräften direkt vor Ort abgefragt werden. Dies spart Weg- und Kommunikationszeiten ein und führt somit zu einer Erhöhung der Arbeitseffizienz.

Ein weiteres Anwendungsgebiet ist der Bereich der Fernwartung, die in Zukunft mit der Verfügbarkeit schneller Mobilfunknetze (UMTS, GPRS) auch mobil ausgeführt werden könnte. Ein Ingenieur könnte dann von unterwegs mit einem PocketPC (z.B. Pocket LOOX /2/) auf die Leitwartendaten zugreifen.

Eine besondere Rolle bei der Präsentation von Prozeßinformationen spielen Prozeßvisualisierungen. Hierbei werden Prozeßzustände in komplexen graphischen Darstellungen symbolhaft repräsentiert und laufend aktualisiert. Bei der Projektierung einer Anlage werden solche Visualisierungen aufgrund der erforderlichen hohen Informationsdichte in der Regel fest für die Bildschirmauflösung der Displays in der Leitwarte entworfen.

Greift man mit Tablet-Computern bzw. PocketPCs auf diese Bilder zu, so ist die Bildschirmauflösung solcher tragbaren Geräte (meist 800x600 bzw. 320x240 Pixel) geringer als die der Leitwarten-Displays (meist 1024x768 bzw. 1280x1024 Pixel). Damit kann auf dem Mobilrechner nur ein Ausschnitt des Prozeßbildes angezeigt werden, der durch Scrollen verschiebbar ist. Diese Realisierung weist den Nachteil auf, daß der Überblick über die Gesamtdarstellung verlorenght und damit z.B. Alarmzustände übersehen werden können, die in nicht sichtbaren Bereichen dargestellt werden.

Dieses Problem kann durch die Neuprojektierung von Prozeßbildern entsprechend der Auflösungen der Mobilgeräte gelöst werden. Eine solche Vorgehensweise ist jedoch mit hohen Kosten verbunden und eine automatische Anpassung daher wünschenswert. Dabei führt die einfache Skalierung des Bildes in vielen Fällen nicht zum Ziel, weil dadurch Details verlorengehen und somit die Erkennbarkeit der Darstellung leidet.

2 Lösungsansatz

Dieser Beitrag schlägt daher vor, zur Lösung des Auflösungsproblems eine FishEye-View-Technik (vgl. auch /3, 4/) einzusetzen, die hohe Detailliertheit eines Teilbereiches des Bildes (*Fokus*) nahtlos mit einem gestauchten Überblick über die restlichen Bereiche des Prozeßbildes (*Kontext*) verbindet. Damit ist das gesamte Prozeßbild zu jedem Zeitpunkt erkennbar, so daß etwaige Alarmzustände nicht übersehen werden können. Der jeweils wichtige Ausschnitt ist in Originalgröße sichtbar, die restlichen Bildbereiche werden verkleinert dargestellt. Die meisten in der Literatur vorgestellten FishEye-Techniken verwenden im Kontextbereich jedoch eine vom Fokus zum Bildrand hin kontinuierlich zunehmende Stauchung. Dies erfordert einen hohen Rechenaufwand und ist somit für mobile Endgeräte nicht geeignet.

Daher soll hier eine spezielle Form der FishEye-Darstellung, der sogenannte *Rechteckige FishEye-Views* /5/, zur Auflösungsanpassung von Prozeßbildern vorgeschlagen werden. Diese Technik nutzt Zweierpotenzen als Skalierungsfaktoren und unterteilt das Bild in rechteckige Bereiche gleicher Skalierung. Dadurch ist sie schnell berechenbar und somit auch für mobile Geräte geeignet.

3 Der Rechteckige FishEye-View

Abbildung 2 illustriert das Prinzip der vorgeschlagenen Darstellungstechnik. Es wird ein *Fokusbereich* definiert, der in Originalgröße abgebildet wird. Der Fokus ist von sogenannten *Kontextringen* umgeben, die die verbleibenden Teile der Darstellung zwischen Fokusrand und Bildrand derart stauchen, daß das so umgerechnete Prozeßbild genau die verfügbare Displayfläche auf dem Terminal ausfüllt. Jedem Kontextring ist eine Zweierpotenz als Stauchungsfaktor zugeordnet, die vom Fokus zum Bildrand hin wächst.

Ringe mit stärkerer Stauchung ermöglichen eine größere Platzeinsparung als Ringe mit geringerer Stauchung. Neben der Fokusgröße bestimmt somit das Breitenverhältnis der verschiedenen Ringe den Platzbedarf. Durch Wahl geeigneter Ringbreiten ist es somit möglich, die Größe der Darstellung desselben Prozeßbildes an verschiedene mobile Displays anzupassen. Um ein Prozeßbild in XGA-Auflösung (1024x768 Pixel) bildschirmfüllend darzustellen, reicht auf einem Tablet-Computer mit SVGA-Auflösung (800x600 Pixel) ein Kontextring der Stauchung 1/2 aus (siehe Abbildung 3). Für die Anpassung desselben Prozeßbildes an einen Pocket-PC mit QVGA-Auflösung (320x240 Pixel) werden drei Kontextringe mit den Stauchungsfaktoren 1/2, 1/4 und 1/8 benötigt (siehe Abbildung 4). Als günstiger Wert für die Fokusgröße hat sich dabei ein Viertel der Bildschirmfläche erwiesen.

Die Auflösungsanpassung mit dem FishEye-View erfolgt *dynamisch* und *interessengesteuert*. Der Benutzer erwartet den Fokus immer dort, wo aktuell sein größtes Interesse liegt. Dies wird bei Stiftbedienung dadurch realisiert, daß der Fokus bei Bewegungen des Stiftes immer der aktuellen Position des Mausursors nachgeführt wird. Eine solche Nachführung kann – je nach verfügbarer Rechenleistung und Nutzerpräferenzen – sprunghaft oder animiert erfolgen. Dadurch integriert sich der FishEye-View nahtlos in den Interaktionsablauf. Der bedienbare Bereich eines Prozeßbildes kann auf den Fokus eingeschränkt werden. Das bedeutet, daß nur Stifteingaben im Fokus als Bedienoperationen interpretiert werden. Stiftaktionen außerhalb des Fokus werden als Veränderung des Nutzerinteresses gewertet, und der Fokus wird automatisch entsprechend nachgeführt. Durch diese dynamische Fokuspositionierung wird so der jeweils relevante Teil des Prozeßbildes in voller Auflösung dargestellt.

Damit der Nutzer die Aufteilung der Darstellung in Fokus und Kontext nachvollziehen kann, muß der Fokusbereich visualisiert werden. Dafür kommen zwei verschiedene Möglichkeiten in Betracht. Der Fokusbereich kann umrahmt dargestellt werden (siehe Abbildung 4), oder ein Farbattribut des Fokus bzw. des Kontextes kann geändert werden. Abbildung 3 zeigt exemplarisch eine Aufhellung des Fokusbereiches, ebenso kann z.B. der Kontextbereich abgedunkelt oder die Farbsättigung im Kontextbereich verringert werden.

4 Realisierung im Kontext einer mobilen Leitwarte

Der mobile Zugriff auf Leitwartendaten erfolgt i.d.R. über Thin-Client-Lösungen. Zu nennen sind hierbei vor allem die Microsoft Terminal Services, das System Citrix Metaframe und das frei verfügbare plattformübergreifende Fernsteuersystem VNC /6/. Diese Systeme bestehen aus einer Serverkomponente auf dem Leitwarten-PC, die die Bildschirmausgaben der Anwendungen an das mobile Gerät weiterleitet, und einer Clientkomponente auf dem mobilen Gerät, die diese Bildschirmausgaben darstellt und Benutzereingaben an den Server zurückübermittelt. Das mobile Gerät verfügt somit zu jedem Zeitpunkt über eine Kopie des Bildschirminhaltes auf dem Server, weshalb diese Systemarchitektur auch als „entfernter Bildschirmspeicher“ (Remote Framebuffer) bezeichnet wird. Eine solche Thin-Client-Lösung kann prinzipiell beliebige Anwendungen steuern – im Fall einer mobilen Leitwarte handelt es sich bei der gesteuerten Anwendung um ein Prozeßvisualisierungssystem.

Wird der FishEye-View in ein Thin-Client-System zum mobilen Zugriff auf Prozeßinformation integriert, arbeiten Client und Server mit verschiedenen Bildschirmkoordinatensystemen. Somit ist eine *FishEye-Transformation* sowohl des ausgegebenen Bildes als auch der graphischen Eingaben erforderlich. Das Ausgabebild im Remote Framebuffer wird hierzu in Teil-

rechtecke unterteilt, die unterschiedlich stark skaliert und im Bildschirmkoordinatensystem des Clients zusammengesetzt werden. Eine Stifteingabe im auf dem Client wird analog durch Anwenden der umgekehrten Vorschrift in das Koordinatensystem des Servers umgerechnet. Dazu wird in die Client-Software auf dem mobilen Endgerät eine FishEye-Komponente zwischengeschaltet, die die Umrechnung der graphischen Ein- und Ausgaben vornimmt sowie den der FishEye-Transformation unterworfenen Bildschirminhalt darstellt. Außerdem behandelt diese Komponente die *lokalen Interaktionen*, die lediglich zu einer Verschiebung des Fokus führen, aber keine Eingaben für die auf dem Server laufende Software darstellen.

5 Ausblick

Zukünftige Prozeßvisualisierungssysteme werden sehr stark mit der Problematik heterogener Endgeräte konfrontiert werden, die eine Adaptivität von Prozeßbildern erfordern. Daher erscheint es sinnvoll, bei der Konzeption solcher Systeme „eingebaute“ Techniken zur Adaptation anzubieten, die von neuen Prozeßvisualisierungsprojekten genutzt werden können. Hierbei stellt der FishEye-View ein mögliches Werkzeug zur Auflösungsanpassung von Pixelgrafiken bzw. zur nicht-uniformen interessenabhängigen Skalierung von Vektorgrafiken dar, neben anderen Werkzeugen wie z.B. Zoomwerkzeugen und flexiblen Layout-Managern. Aus einem solchen Werkzeugkasten kann dann für jede Aufgabe das passende Werkzeug gewählt werden. Beispielsweise könnte bei der Visualisierung eines Rohrleitungssystems ein Zoom-Manager zum Einsatz kommen, der ein Hineinzoomen in das Leitungsnetz von einem globalen Überblick („World View“) bis hin zur detaillierten Darstellung eines defekten Ventils ermöglicht. Durch die Kombination des Zoom-Managers mit dem FishEye-View kann auch die Umgebung des defekten Ventils im Überblick dargestellt werden. Für die Anzeige der Bedienelemente des Programms (z.B. die Buttonleiste) ist der FishEye-View weniger gut geeignet. Dazu könnte ein Layout-Manager zum Einsatz kommen, der die Anpassung auf der Grundlage anderer Regeln (z.B. durch Verringern des Abstandes der Buttons) vornimmt.

Eine prototypische Integration des FishEye-Views in eine Thin-Client-Lösung auf der Basis von VNC zeigte, daß das vorgeschlagene FishEye-Konzept praktikabel ist. Besonders hervorzuheben ist, daß es sich hierbei um eine generische Lösung handelt, d.h., daß sie in existierende Prozeßvisualisierungsumgebungen eingebettet werden kann, ohne daß eine Neuprojektierung von Bildern erforderlich ist. Eine solche generische Lösung ist aus ergonomischer Sicht manchmal suboptimal, da der gesamte Bildschirm- bzw. Fensterinhalt der FishEye-Transformation unterworfen ist, obwohl bestimmte Elemente des Nutzerinterfaces besser nach anderen Regeln angepaßt werden sollten.

Die vorgestellte FishEye-Technik als Thin-Client-Lösung bietet auf dem Weg zu den oben beschriebenen adaptiven Systemen eine Migrationslösung, die es ermöglicht, existierende Prozeßbilder auf mobilen Endgeräten platzsparend darzustellen.

6 Referenzen

- /1/ Siemens AG: MOBIC Website. <http://www.ad.siemens.de/mobic/>.
- /2/ Fujitsu Siemens Computers: PocketLOOX Website. [http:// http://www.fujitsu-siemens.com/PocketLOOX](http://www.fujitsu-siemens.com/PocketLOOX).

- /3/ Zinser, K.: Fisheye Views - Interaktive, dynamische Visualisierungen. In: atp -Automatisierungstechnische Praxis, 37 (9), S. 42-50, 1995.
- /4/ Sarkar, M. and Brown, M.H.: Graphical fisheye views of graphs. Technical Report No. 84, digital Systems Research Center, 1992.
- /5/ Rauschenbach, U.: The Rectangular Fish Eye View as an Efficient Method for the Transmission and Display of Large Images. In: Proceedings of IEEE ICIP'99, Kobe, Japan, Oct. 25-28, 1999.
- /6/ Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood & Andy Hopper: Virtual Network Computing, IEEE Internet Computing, Vol. 2, No. 1, Jan/Feb 1998, pp33-38.

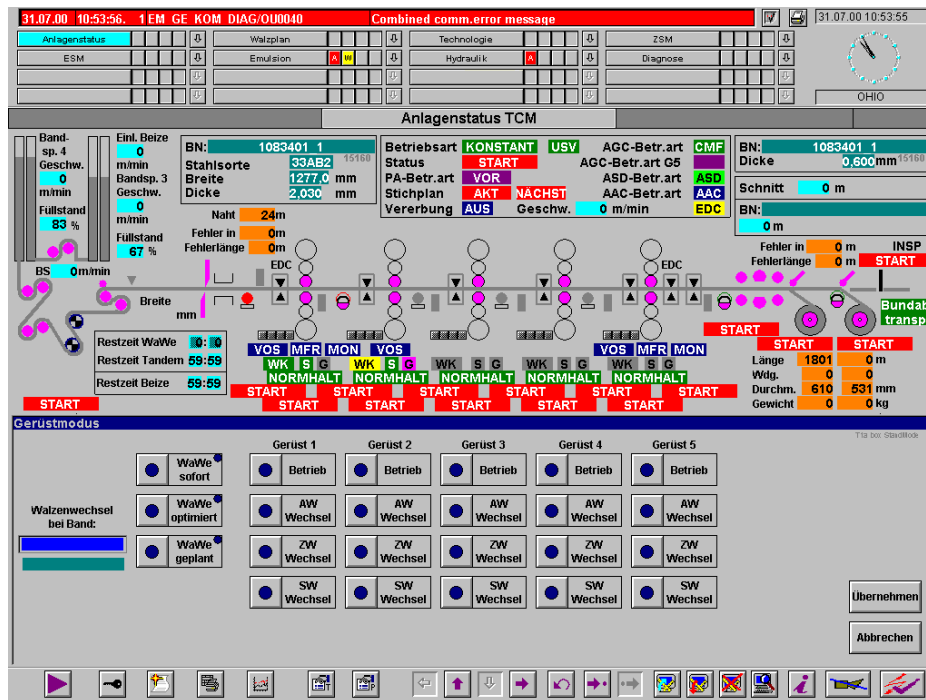


Abbildung 1: Ein Prozeßbild der Größe 1024x768 Pixel.

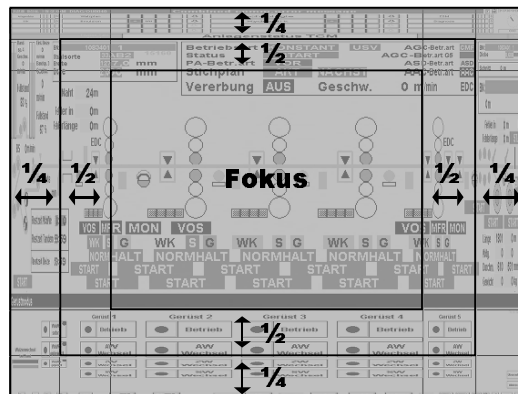


Abbildung 2: Skalierungsprinzip des Rechteckigen FishEye-View.

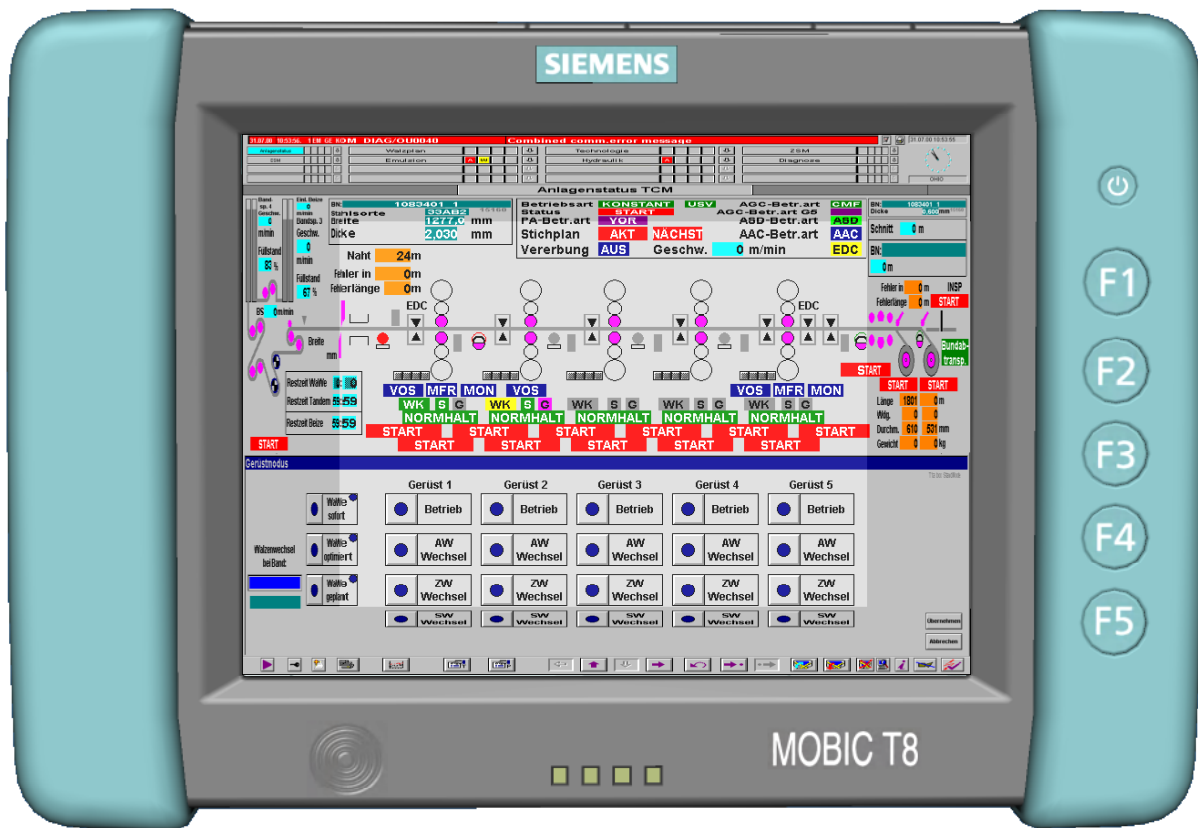


Abbildung 3: FishEye-Darstellung des Prozeßbildes aus Abbildung 1 auf einem robusten mobilen Tablet-Computer. Auflösungsreduktion von 1024x768 auf 800x600 Pixel (Platzeinsparung 39%). Die Fokusregion ist heller dargestellt.

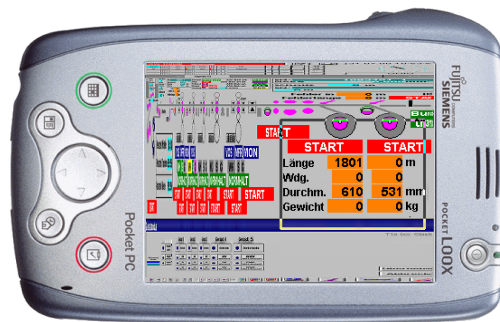


Abbildung 4: FishEye-Darstellung des Prozeßbildes aus Abbildung 1 auf einem PocketPC. Auflösungsreduktion von 1024x768 auf 320x240 Pixel (Platzeinsparung 90%). Die Fokusregion ist umrahmt dargestellt.