

AMBIENT INTELLIGENCE - VISION UND TECHNISCHE LÖSUNGSANSÄTZE

Christian Hentschel

*Brandenburgische Technische Universität Cottbus
christian.hentschel@tu-cottbus.de*

Abstract: „Ambient Intelligence“ umschreibt Szenarien, bei der die Endnutzer im Mittelpunkt stehen und die Technik nahezu unsichtbar im Hintergrund agiert. Technische Aspekte sind die Vernetzung im Haus, intelligente Benutzerschnittstellen mit zentralen und dezentralen Sensoren, Bild, Sprache und vieles mehr. Neben der skalierbaren Übertragung (Quality-of-Service) werden neuartige, skalierbare Consumer Terminals, die sich dynamisch an die Inhalte und vorhandenen Hardware-Kapazitäten anpassen, immer wichtiger. Eine erste Implementierung eines skalierbaren Consumer Systems und die erzielten Eigenschaften werden vorgestellt.

1 Einführung und Aspekte einer Ambient Intelligence Umgebung

Ein sehr aktuelles, breites Forschungsgebiet kann mit „Ambient Intelligence“ (AmI) umschrieben werden, bei der die Endnutzer im Mittelpunkt stehen und die Technik nahezu unsichtbar als „Butler“ dient. Die Technik stellt sich auf die Wünsche des Nutzers ein und nicht umgekehrt.

In [1] werden unterschiedlichste Aspekte von Ambient Intelligence betrachtet und eine Reihe beispielhafter Anwendungen vorgestellt. An dieser Stelle kann nicht umfassend auf das komplexe Thema eingegangen werden, jedoch sollen mögliche Szenarien beispielhaft aufgezeigt werden. Eine oft gestellte Frage ist der Sinn und Nutzen neuer Applikationen, und es gibt genügend Beispiele von Entwicklungen, die an den Bedürfnissen der Consumer vorbei gehen. Vielleicht sollte die Frage anders formuliert werden: Wie ist meine persönliche Situation, was möchte ich erreichen und welche Schwierigkeiten sind mit der Realisierung verbunden? Jedem fällt sofort das Beispiel eines Videorekorders ein, bei dem in Abwesenheit des Nutzers eine Sendung aufgezeichnet werden soll. Oft scheitert die Aufzeichnung an der schwierigen oder falschen Bedienung des Gerätes, oder am nicht unerheblichen Zeitaufwand zur Programmierung. Features und im ungünstigsten Fall Geräte, die technisch zwar vorhanden, aber nicht bedienbar oder bekannt sind, nützen dem Consumer wenig. Ziel ist folglich ein intuitives Verstehen der Funktionen und Möglichkeiten sowie eine extrem einfache, intuitive Bedienung.

Die Mensch-Maschine-Schnittstelle ist von zentraler Bedeutung. Die heutigen, vielfach vorhandenen Fernbedienungen sind ohne funktionelle, zum Teil technische Kenntnisse, von vielen Anwendern nur schwer nutzbar. Ziel muss es daher sein, nicht das Maschineninterface einfach zu halten, sondern die natürliche Kommunikation (Sprache, Gestik) und Umgebungsparameter (anwesender Nutzer, Zeit, Beleuchtung, Temperatur usw.) als Standard zu nehmen. Die Umgebungsparameter können zur komplexen Situationsanalyse und automatischen Steuerung genutzt werden, ohne Eingriffe durch den Nutzer zu verlangen. Bei Abwesenheit eines Zuhörers werden zum Beispiel Fernseher und Radio automatisch abgeschaltet, oder in einer besonderen (Not-)Situation (überlaufender Topf auf dem Herd, Rauchentwicklung, Schlagregen bei offenem Fenster) die entsprechenden Maßnahmen ergriffen.

Ein weiterer Aspekt ist die Vernetzung der Geräte. Was nützt mir ein DVD-Player in meinem Desktop-PC im Arbeitszimmer, wenn ich eine DVD auf dem Fernseher im Wohnzimmer wiedergeben möchte? Die derzeitige Lösung ist das Vorhandensein vieler Geräte mit ähnlichen oder gleichartigen Funktionen, die auch noch unterschiedliche Benutzerschnittstellen

haben. Die Vernetzung im PC-Bereich breitet sich inzwischen auch auf den Heimbereich aus, die Vernetzung von Consumergeräten steckt dagegen noch in den Startlöchern. Ein weiteres praktisches Beispiel ist eine Routenplanung für eine längere Autofahrt. Die Planung mit Fahrt und Tagesablauf findet in der Wohnung und nicht im Auto statt, jedoch müssen die Informationen (Routenführung, zu besichtigende Plätze, Restaurantaufenthalte, Hotel) während der Fahrt verfügbar sein, ohne den Fahrer von seiner eigentlichen Aufgabe, dem Steuern des Fahrzeugs, abzulenken. Eine automatische Übertragung der Informationen ohne aktives Eingreifen des Nutzers wäre hilfreich und zeitsparend.

Bei der persönlichen Situation zu Hause entscheiden viele Aspekte über den Sinn und Nutzen von „unsichtbaren“ Helfern im Hintergrund. Aspekte sind die persönliche gesundheitliche Situation, das Alter, die weiteren Bewohner, spezielle Bedürfnisse in Bezug auf Komfort, Sicherheit und Kommunikation usw. Was für einen Nutzer eine große Hilfe darstellt (z.B. sind alle Fenster beim Verlassen der Wohnung geschlossen?), hat für andere Nutzer gar keinen Wert.

Es gibt zahllose Beispiele für „kleine Helferlein“, von denen nur zwei Möglichkeiten (Spiegel und Fenster) angedeutet werden sollen. Zu betonen ist, dass diese Möglichkeiten schon jetzt oder in naher Zukunft technisch machbar sind. Morgens im Bad verbringt man einige Zeit vor dem Spiegel. Je nach Nutzer könnten in einem Display, das in einen Bereich des Spiegels sichtbar werden kann, die neuesten Nachrichten, Wetter, Börsendaten, Post oder Terminkalender wiedergegeben werden. Selbst die Lautsprecher sind möglicher Bestandteil des Spiegels. Die wiederzugebenden Inhalte könnten individuell an dem gerade vor dem Spiegel stehenden Nutzer angepasst werden. Ein Fenster als zweites Beispiel dient in erster Linie als natürlicher Lichtzugang, jedoch ergeben sich je nach Situation andere Anforderungen. Es könnte alternativ als Großdisplay für verschiedene, auch interaktive Zwecke verwendet werden oder als praktischer, elektronischer Sonnenschutz dienen. Hier kann die Synergie zwischen verschiedenen Anwendungen vorteilhaft genutzt werden.

Ambient Intelligence bezieht sich auf eine elektronische Umgebung, die sensitiv und eigenständig auf die Anwesenheit und Bedürfnisse von Personen reagiert. AmI kombiniert allgegenwärtige Signalverarbeitung mit intelligenten Mensch-Maschine-Schnittstellen. Technische Aspekte sind die Vernetzung im Haus, intelligente Benutzerschnittstellen mit zentralen und dezentralen Sensoren, Bild, Sprache und vieles mehr.

Eine Vernetzung betrifft interne multimediale Consumer Terminals, PCs und Haushaltseinrichtungen sowie Zugang zu externen Service-Netzwerken wie IP, TV und Teleservices. Gegen eine zentrale Steuerung im Heimbereich spricht die Störanfälligkeit und schlechte Erweiterbarkeit mit neuen, dezentralen Komponenten. Ein modulares System hat den Vorteil der individuellen Anpassung und evolutionären Einführung. Neben verkabelten Lösungen mit entsprechendem Aufwand bieten sich kabellose Verbindungen an, die derzeit in ihrer Bandbreite und Störsicherheit jedoch nicht immer den Ansprüchen genügen.

Ein flexibles System, wie es eine Ambient Intelligence Umgebung erfordert, sollte Anwendungen (Applikationen) als Ganzes erfassen und sowohl die Netzwerkressourcen als auch die Terminalressourcen mit einbeziehen.

2 Flexibilität in Netzwerken und Consumer Terminals durch Quality-of-Service

Verglichen mit Multimedia-Applikationen in PC-Netzwerken haben Applikationen in Consumer Terminals (TVs, Set-Top Boxen, usw.) andere Anforderungen. Consumer Terminals erfordern Echtzeitverarbeitung und Robustheit gegenüber Störungen. Gerade Robustheit wird allgemein vorausgesetzt, ein erneutes „Hochfahren“ eines Fernsehers oder CD-Spielers ist

nicht akzeptabel. Die Formate für Video, Audio und Grafik unterscheiden sich zum Teil erheblich von denen in PC-Netzwerken.

Die Vernetzung und parallele Übertragung von Multimedia-Daten (Video, Audio, Grafik Text) begann mit PC-Netzwerken. Aus Sicht eines einzelnen Nutzers genügt die Datenübertragung als Punkt-zu-Punkt Verbindung über drahtgebundene oder drahtlose Kanäle. Bei höherer Netzwerkbelastung durch eine Vielzahl von Nutzern entstehen nicht akzeptable Verzögerungen oder Zugriffsabweisungen. Die problematische Größe ist die begrenzte Übertragungsbandbreite oder Übertragungsrate. Quality-of-Service wurde für Netzwerke eingeführt, um diese Übertragungsprobleme zu lösen und jedem Nutzer nach Priorität Kapazitäten zuzuweisen. Typische QoS Parameter in Netzwerken für Video Streaming sind Bildauflösung, Bildfrequenz, Farbauflösung und Qualität der Kompression, um die Übertragungsbandbreite zu reduzieren. Zusammengefasst ergibt sich eine Optimierung zwischen der Übertragungsbandbreite und der Qualität der Applikation.

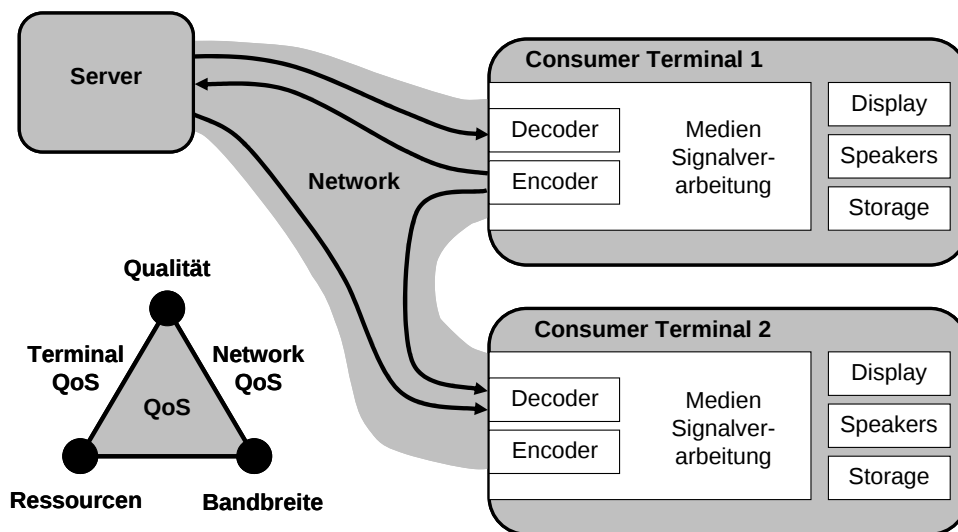


Abbildung 1 – Unterschiede zwischen Terminal-QoS und Netzwerk-QoS

Abbildung 1 zeigt zwei zukünftige Consumer Terminals und einen Server innerhalb eines Netzwerks. Die Consumer Terminals besitzen einen Decoder und optional einen Encoder als Netzwerk Interface, die erforderliche Signalverarbeitung und audio-visuelle Ausgänge (Display, Speakers) oder Speichermöglichkeiten. Die Medien-Signalverarbeitung ist abhängig von den gewählten Eingangsquellen, ihrer Qualität und der Leistungsfähigkeit der Consumer Terminals. Typischer Weise sind die Ressourcen zur Medien-Signalverarbeitung begrenzt, so dass hier nicht zwischen Übertragungsbandbreite und Qualität, sondern zwischen Ressourcen und Qualität optimiert werden muss. Audiounterbrechungen führen zu einer sehr schlechten Qualität, Einzelbildwiederholungen bei Video-Streaming werden dagegen weniger schlecht bewertet.

Fasst man beide QoS Optimierungen zusammen, ergibt sich das Dreieck in Abbildung 1 mit den Eckpunkten Qualität – Ressourcen – Bandbreite. Als Gesamtapplikation z.B. einer MPEG-2 Übertragung beeinflussen sich alle Parameter gegenseitig mit der Schwierigkeit der Optimierung über das gesamte System.

Neben der skalierbaren Übertragung (Quality-of-Service), die bereits in der PC-Welt ein etabliertes Forschungsgebiet darstellt, wurden eigene Untersuchungen für neuartige, skalierbare Consumer Terminals durchgeführt. Diese Systeme passen sich dynamisch an die Inhalte und vorhandenen Hardware-Kapazitäten an. Zukünftige Consumer Terminals (TVs, Set-Top Boxen, Displays und mobile Geräte) werden hochwertige Video- und Audioapplikationen mit Multimediaanwendungen aus dem PC Bereich verbinden. Die Basis bilden programmierbare

Plattformen, die Signalverarbeitung verschiedener Medien in Software erlauben. Die erwarteten Vorteile sind unter anderem flexible, zukunftssichere und offene, erweiterbare Produkte. Eine erste Implementierung eines skalierbaren Systems und die erzielten Eigenschaften werden vorgestellt.

Programmierbare Komponenten sind speziell im Videobereich sehr kostspielig und verbrauchen derzeit mehr Energie im Vergleich zu funktionspezifischen Realisierungen. Dazu kommt, dass Consumer Terminals aus Kostengründen stark in ihren Ressourcen begrenzt sind. Wir zeigen, dass diese Nachteile durch Skalierbare Media Algorithmen (SMAs) überwunden werden, die die programmierbaren Komponenten sehr effizient und flexibel nutzen können.

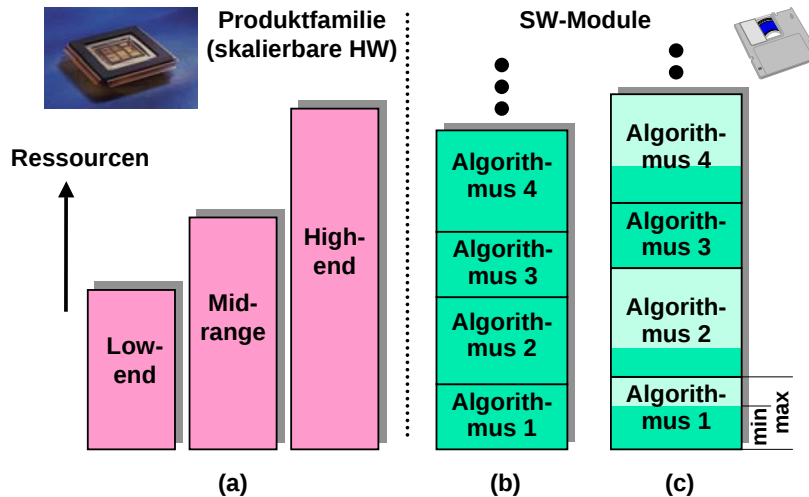


Abbildung 2 - Programmierbare Produktfamilie und feste oder skalierbare Softwarealgorithmen

Abbildung 2 zeigt eine Produktfamilie auf der Basis programmierbarer Komponenten mit unterschiedlicher Leistungsfähigkeit in Verbindung mit traditionellen oder skalierbaren Softwarealgorithmen. Plattformen mit unterschiedlicher Leistungsfähigkeit werden nebeneinander für unterschiedliche Marktsegmente existieren. Derzeitige Algorithmen zur Signalverarbeitung sind für die höchste Qualität bei gegebenen Ressourcen konzipiert. In Abbildung 2b illustriert die Höhe der Algorithmen die benötigten Ressourcen zur Signalverarbeitung. Die benötigten Ressourcen und die Qualität am Ausgang sind normalerweise nicht skalierbar. Die Anzahl der gleichzeitig laufenden Algorithmen ist somit stark von der Leistungsfähigkeit der Plattform abhängig und begrenzt. Diese Grenze kann durch SMAs überwunden werden (Abbildung 2c). SMAs haben typischerweise eine Kernfunktion mit minimalen Ressourcenanforderungen, aber auch minimaler Ausgangsqualität (dunkle Bereiche). Die Ausgangsqualität kann graduell gesteigert werden unter Verwendung weiterer Ressourcen (helle Bereiche).

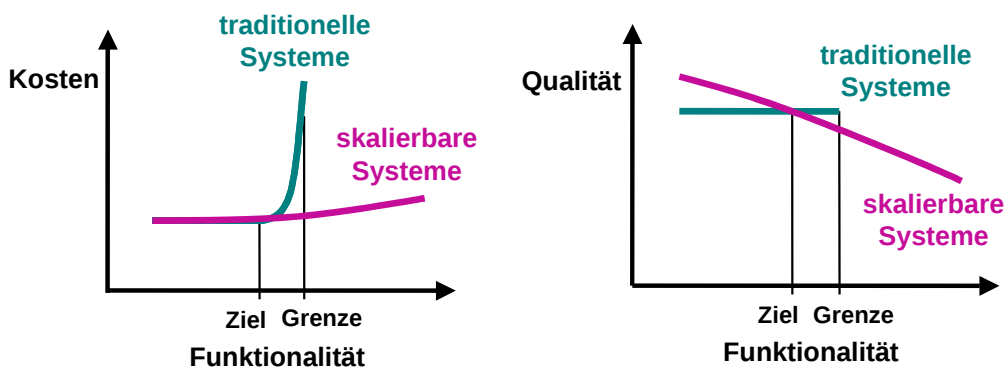


Abbildung 3 - Überwindung der Grenzen heutiger programmierbarer Plattformen

Traditionelle Systeme werden für eine bestimmte Funktionalität bei hoher Qualität entwickelt (Abbildung 3). Die Funktionalität kann nur wenig über die Zielfunktionalität gesteigert werden, und dies zu hohen Kosten. Ein skalierbares System hat keine derartigen Grenzen, stattdessen ist die Qualität abhängig von der gleichzeitig genutzten Funktionalität. Sogar nachträglich implementierte Applikationen können hohen Qualitätsanforderungen genügen, solange nicht zu viele Applikationen zur gleichen Zeit laufen. Diese Flexibilität ist insbesondere für zukünftige Multimediaapplikationen erwünscht und ermöglicht unter anderem eine schnelle Markteinführung neuer Applikationen oder Systemerweiterungen (Upgrades). Die Startpunkte für die Kosten-Funktionalitätskurven wurden willkürlich gleich angesetzt, da sich bei einem skalierbaren System sowohl Argumente für höhere Initialkosten (Silizium, Systementwicklung), als auch für geringere Betriebskosten (Nutzung auf verschiedenen Plattformen, erweiterte Verwendung der Hardware und Software, Funktionstests) finden.

SMAs sind Algorithmen, die sich dynamisch in ihrer Ausgangsqualität in Relation zu den verfügbaren Ressourcen steuern lassen. Diese dynamische Kontrolle der Qualität und genutzten Ressourcen ist in traditionellen Systemen unbekannt. Weiterhin müssen Softwarelösungen in Consumer Terminals stabile, robuste und kosteneffektive Systeme garantieren. Ein speziell für Consumer Terminals entwickelter Quality-of-Service Resource Manager (QoS-RM) unterstützt die dynamische Steuerung.

Der QoS-RM weist die zur Verfügung stehenden Ressourcen den SMAs zu und optimiert während des Betriebs die Qualität des Gesamtsystems. SMAs und QoS-RM unterstützen offene Systeme in einer sich schnell weiterentwickelnden Multimedialandschaft [2-8].

Zusammenfassend werden für Consumer Terminals, die auf SMAs mit dynamischer Ressourcenverwaltung basieren, folgende Vorteile gegenüber traditionellen System angestrebt:

- Flexibilität,
- zukunftssichere und aufrüstbare Produkte,
- kurze Zeit bis zur Markteinführung neuer Applikationen,
- Wiederverwendung von Softwaremodulen um Produktfamilien zu unterstützen, und
- Kosteneffizienz.

Im folgenden Kapitel wird das erste Echtzeitsystem vorgestellt, das auf diesen Ideen basiert. Die untersuchten Ziele betreffen die Realisierbarkeit und ein Nachweis der Vorteile dieses neuartigen Systems.

3 Realisierung eines skalierbaren Consumer Terminals

3.1 Skalierbare Videoalgorithmen

Ein auf Ressourcen-Qualität skalierbarer Algorithmus für Consumer Terminals muss von einem auf Bandbreite-Qualität skalierbaren Algorithmus unterschieden werden. Die Letzteren werden in Übertragungsnetzwerken verwendet. Ein Ressourcen-Qualität skalierbarer Algorithmus erlaubt die dynamische Adaption der Ausgangsqualität in Verbindung mit den genutzten Ressourcen auf einer gegebenen Plattform, unterstützt verschiedene Plattformen/Produktfamilien zur Signalverarbeitung, und ist einfach über eine Schnittstelle zu steuern und zu kontrollieren.

Abbildung 4 zeigt die Struktur eines skalierbaren Algorithmus. Im Fall von skalierbaren Videoalgorithmen sprechen wir von SVAs. Das Kontrollinterface ist notwendig für die Kommunikation der zur Verfügung stehenden Qualitätsstufen und den erforderlichen Ressourcen in einer QoS-Umgebung.

Ein Algorithmus kann in spezifische Funktionsblöcke unterteilt werden (Abbildung 4), von denen zumindest ein Block skalierbar ist. Die Eigenschaften des gesamten Algorithmus hängen von der geeigneten Kombination der Qualitätsstufen in den verschiedenen Funktionsblöcken ab. Von allen möglichen Kombinationen sind nur wenige sinnvoll, wie in Abbildung 4 dargestellt ist. Der Block 'Qualitäts-Ressourcen-Kontrolle' beinhaltet diese Information und die geeigneten Einstellungen für die Funktionsblöcke. Die besten Kombinationen sind untereinander als Kurve verbunden und beinhalten die höchste Qualität bei geringstem Ressourcenbedarf.

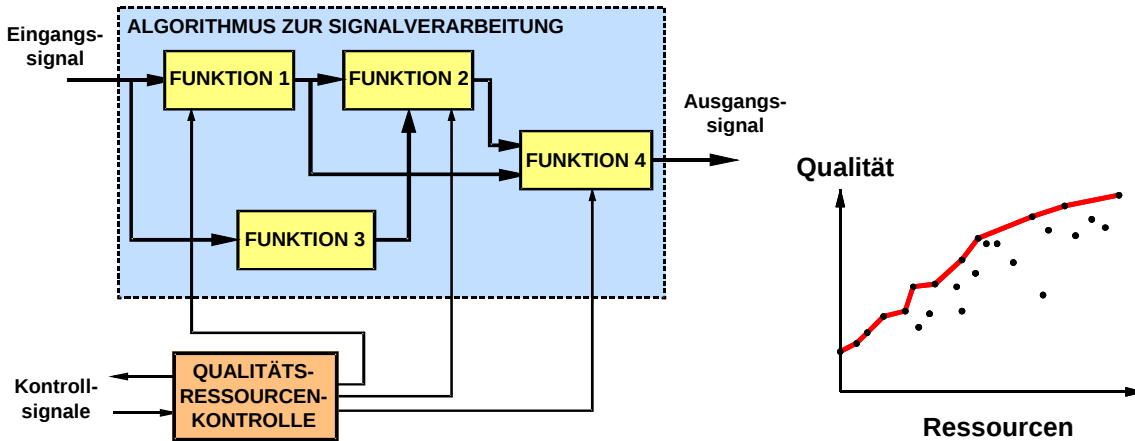


Abbildung 4 - Grundstruktur und Qualitäts-Ressourcen Kombinationen für skalierbare Algorithmen

Eine Voraussetzung für den effektiven Einsatz skalierbarer Algorithmen ist ein großer Regelbereich der Ressourcen, verbunden mit einem geringen Kontrollaufwand. Nur mit diesen Eigenschaften ergeben sich Vorteile gegenüber nicht skalierbaren Algorithmen. Die ersten drei untersuchten Algorithmen sind MPEG-2 Decoder, Bildschärfverbesserung und Scaler für Bild-im-Bild (picture-in-picture, PiP) Anwendungen. Die Algorithmen wurden für eine VLIW (Very Large Instruction Word) CPU optimiert (TM 1300 von TriMedia Technologies Inc.®) und die benötigten Ressourcen gemessen. Für den MPEG-2 Decoder werden im Durchschnitt 75-110 MIPS benötigt. Absolut gesehen ist der Ressourcenbereich groß, allerdings wird im Verhältnis kein spektakuläres Ergebnis erzielt. Dies liegt daran, dass im entwickelten MPEG-2 Decoder lediglich zwei Funktionsblöcke skalierbar sind, namentlich die IDCT und die Bewegungskompensation [4, 7].



Abbildung 5 - Ressourcenbereich zweier skalierbarer Algorithmen

Die Bildschärfanhebung ist im oberen Teil von Abbildung 5 illustriert. Da ein Ausgangsbild bereits ohne Signalverarbeitung verfügbar ist, benötigt die Basisfunktion QS0 keine Ressourcen. Der Algorithmus ist somit 100% skalierbar. Bei QS1 findet lediglich eine horizontale

Bildschärfeanhebung statt, was im Tor deutlich zu sehen ist. Fast horizontale Linien wie die Feldlinien werden nicht angeschärft. Bei dieser einfachen Verarbeitung (ca. 17 MIPS für normale Standardauflösung) wird Bildrauschen leider ebenfalls verstärkt, was im Rasen erkennbar ist. Die aufwendigere Signalverarbeitung (QS4) verwendet zweidimensionale Filter und Schutzmechanismen gegen Rauschen und Clipping. Der Ressourcenbedarf steigt auf 29 MIPS an [8].

Der skalierbare Scaler für PiP Anwendungen reduziert die Größe mit einem Faktor von 4 in beiden Dimensionen. Die Grundfunktion (QS0) wird durch einfache Unterabtastung erreicht. Einfache oder komplexere Filter verbessern die Bildqualität erheblich. Die benötigten Ressourcen belaufen sich auf 4-14 MIPS. Der Kontrollaufwand beträgt bei allen Algorithmen lediglich 1-2% und ist somit vernachlässigbar.

3.2 Systemarchitektur

Die Systemarchitektur entstand in enger Zusammenarbeit zwischen Experten im Videoapplikationsbereich und Experten im Systembereich. Abbildung 6 zeigt die Architektur aus Sicht der Systemkontrolle. Ein Set von modularen SVAs wird zur Signalverarbeitung benötigt, um die verschiedenen Funktionen in einer Set-Top-Box, TV oder Multimedia-PC auszuführen. Der Strategy Manager selektiert und konfiguriert die SVAs und übernimmt die applikationsspezifische Qualitäts-Ressourcenkontrolle. Der Quality Manager ist für die Systemoptimierung verantwortlich und startet und stoppt Applikationen. Der Resource Manager überwacht die Ressourcen und garantiert und vollstreckt Resource Budgets. Wenn Teile einer Applikation zeitlich mehr oder weniger Ressourcen benötigen, können die Resource Budgets entsprechend angepasst werden.

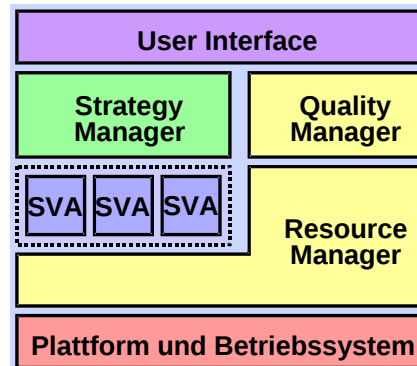


Abbildung 6 - Systemarchitektur, Control Layer

Diese Architektur sieht eine Separierung zwischen der applikationsspezifischen Signalverarbeitung und der Systemsoftware vor. Die applikationsspezifische Signalverarbeitung optimiert die wahrnehmbaren Qualitäten für einen Ressourcenbereich, während die Systemsoftware die tatsächlich zur Verfügung stehenden Ressourcen überwacht und für die Signalverarbeitung garantiert. Die von der Systemsoftware garantierten Ressourcen können allerdings nicht direkt mit von den Applikationen benötigten Ressourcen gleichgesetzt werden, da bei manchen Algorithmen, wie z.B. MPEG Decoder, die benötigten Ressourcen signalabhängig sind. Weiterhin ist das Verhalten programmierbarer Plattformen nicht im Einzelnen vorher bestimmbar, es können z.B. aufgrund von Interrupts, Cache-stalls usw. unvorhergesehene Verzögerungen in der Signalverarbeitung auftreten. Eine bestimmte Qualitätsstufe kann daher nicht bestimmten Ressourcen zugeordnet werden. Stattdessen muss der Stand der Signalverarbeitung gemessen und die Budgets entsprechend dem Bedarf adaptiert werden.

3.3 Kurzzeitige und längerfristige Überlastungen

Um Überlastsituationen vollständig zu vermeiden, müssten für alle Applikationen inklusive der Systemsoftware die Ressourcen für den Worst-case zur Verfügung gestellt werden. In diesem Fall werden die vorhandenen Ressourcen niemals voll genutzt und das System kann niemals kosteneffektiv arbeiten. Das andere Extrem, die durchschnittlich benötigten Ressourcen zur Verfügung zu stellen, ist unrealistisch. Es würde eine ständige Überlastsituation entstehen, die nicht mehr gepuffert werden kann. Also liegt die Kunst darin, den besten Kompromiss zwischen den durchschnittlich und maximal benötigten Ressourcen zu finden, ohne die Robustheit zu kompromittieren.

Abbildung 7 illustriert den imaginären Ressourcenbedarf zweier Applikationen über der Zeit. Die Höhe der Zeichnung entspricht den zur Verfügung stehenden CPU Ressourcen. Applikation 1 konsumiert in der Zeichnung die Ressourcen von unten aufwärts und Applikation 2 von oben abwärts. Die festgelegten Budgets 1 und 2 sind die garantierten Budgets für die beiden Applikationen. In dieser Illustration entspricht die Summe der garantierten Budgets den gesamten CPU Ressourcen. Im praktischen Betrieb sollte die Summe kleiner sein, um auf Überlastsituationen flexibler reagieren zu können.

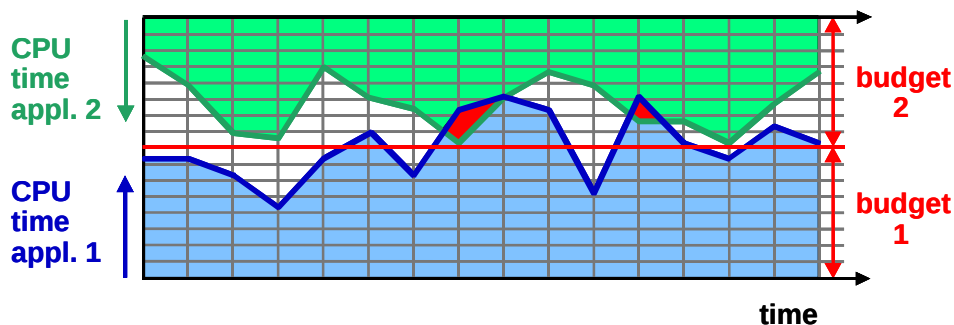


Abbildung 7 - Kurzzeitige Überlastung bei garantierten Budgets

Die Systemsoftware (Quality Manager und Resource Manager) ist für die Zuteilung der Budgets und deren Durchsetzung verantwortlich. Die garantierten Budgets für die Applikationen sind notwendig, um zeitliche Interferenzen zu lösen. Ohne die garantierten Budgets könnte eine eventuell fehlerhafte Applikation alle CPU Ressourcen beanspruchen und damit alle anderen Applikationen oder das Gesamtsystem blockieren. Angemerkt sei, dass zeitliche Interferenzen ein wesentlicher Grund für Probleme mit der Systemrobustheit und unvorhersehbaren Verhalten auf PCs und Workstations sind.

Applikation 1 überschreitet in Abbildung 7 an vier Stellen das garantierte Budget, während Applikation 2 während des dargestellten Zeitraums immer im garantierten Bereich bleibt. In zwei der vier Fälle stehen genügend ungenutzte Ressourcen zur Verfügung, um beide Applikationen störungsfrei laufen zu lassen. In zwei Situationen dagegen kommt es zum Konflikt, dargestellt durch dunkel schattierte Bereiche. In diesem Fall erhält Applikation 2 die Ressourcen, die für diese Applikation garantiert sind. Applikation 1 muss dagegen mit entsprechend weniger Ressourcen auskommen. Bei diesen zeitweiligen Konflikten wird sichergestellt, dass die nicht beteiligten Applikationen ungestört weiter laufen können.

Längerfristige Überlastungen können bei Applikationen entstehen, bei denen sich die benötigten Ressourcen z.B. auf Grund eines Szenenwechsels ändern. Diese längerfristige Überlastung wird zunächst detektiert, was auf Grund der Mittelwertbildung einige Zeit in Anspruch nimmt. Nach der Detektierung muss die Ressourcenverteilung neu bestimmt und die garantierten Budgets entsprechend angepasst werden.

3.4 Implementierung und Ergebnisse

Zur Implementierung des Systems verwendeten wir ein DSP Board mit einem VLIW Prozessor TM 1300 @ 180 MHz, das in einem PC installiert wurde. Das System mit dynamischer Ressourcenverwaltung ist in Abbildung 8 dargestellt. Das System unterstützt einen digitalen Eingang und zusätzlich einen analogen Eingang, beide mit Standardauflösung. Die Audio- und Videoausgänge sind mit einem Fernseher verbunden, der lediglich als Monitor ohne eigene Signalverarbeitung dient. Die gesamte Signalverarbeitung und die Systemsoftware laufen auf dem DSP Board.

Die Echtzeimplementierung auf einem VLIW-DSP beinhaltet skalierbare Algorithmen (MPEG-2 Video Decoder, Kantenanschräpfung, PiP), nichtskalierbare Algorithmen (Demultiplexer, Audio Decoder, Software Mixer, Hardware Mixer, MPEG-1 Encoder) und Kontrollmodule (Strategy Manager, QoS Resource Manager, Grafikeinblendung mit Leistungsmessung).

Verschiedene Applikationen und Modi sind implementiert und über eine Infrarot-Fernbedienung steuerbar sollen aber nicht alle beschrieben werden. Mode 1 ermöglicht MPEG-2 Decodierung für Video und Audio plus Kantenversteilerung in voller Auflösung. Das On-Screen Display ist in allen Modi aktiv und zeigt die selektierten Qualitätsstufen und die aktuell genutzten Ressourcen der drei skalierbaren Algorithmen. Diese sind in Abbildung 8 durch einen Pfeil gekennzeichnet. Die insgesamt genutzten Ressourcen der VLIW-CPU werden auf einem separaten Graph angezeigt. In Mode 1 können alle Algorithmen in höchster Qualität laufen, da die CPU nur zu 60-80% genutzt wird.

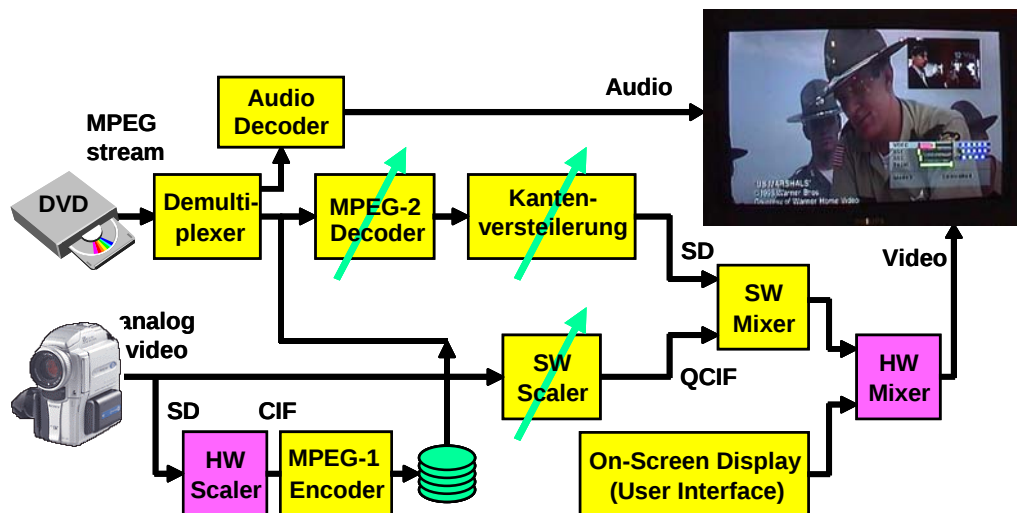


Abbildung 8 - Implementierung des Systems auf einem Multimedia Prozessor

Bedeutender für den Systemtest ist Mode 3. Alle Funktionen von Mode 1 werden unterstützt, und zusätzlich der analoge Eingang mit dem Scaler und Mixer für PiP. Gleichzeitig wird das analoge Eingangssignal im MPEG-1 Format codiert und auf die Festplatte aufgezeichnet. Die benötigten Ressourcen würden bei höchster Qualität die Kapazität der CPU erheblich übersteigen, eine automatische Verringerung der Qualitätsstufen ist somit erforderlich. Aufgrund des datenabhängigen Ressourcenbedarfs kann es sowohl zu kurzzeitigen als auch zu langfristigen Überlastungen kommen.

Eine große Frage war die Systemstabilität bei Überlastsituationen. Im untersuchten System stürzten weder Applikationen ab, noch mussten einzelne Applikationen abgeschaltet werden. Sogar die manuelle Wahl höherer Qualitätsstufen führte nicht zu einem instabilen System. Natürlich konnten manche Applikationen nicht mehr fehlerfrei laufen, so wurden z.B. kurzfristige Audiounterbrechungen hörbar. Mode 4 ist ein Wiedergabemodus für das im Hinter-

grund aufgezeichnete Videosignal. Dieser Mode stellt keine besonderen Anforderungen an die CPU Ressourcen.

Die zusätzlich benötigten Ressourcen für die skalierbaren Algorithmen und die in diesem Fall einfache Kontrollsoftware betragen weniger als 3%. Verglichen mit dem breiten Ressourcenbereich der skalierbaren Algorithmen und den Systemvorteilen erscheint dieser zusätzliche Ressourcenbedarf als geringfügig.

4 Zusammenfassung

Ambient Intelligence eröffnet ein weites Forschungsgebiet, bei der die Applikationen, Situations- und Umgebungsparameter sowie die Mensch-Maschine-Schnittstellen als ganzes betrachtet werden. Neue technologische Ansätze in Bezug auf Sensorik, skalierbare Netzwerke und flexible Consumer Terminals sind notwendig, um evolutionär die visionären Ansätze realisieren zu können.

In Bezug auf zukünftige, flexible Consumer Terminals wurden skalierbare Algorithmen in Kombination mit dynamischer Ressourcenverwaltung vorgestellt. Das neuartige Systemkonzept ermöglicht komplexe Applikationen auch auf DSPs mit begrenzten Ressourcen. Es konnte nachgewiesen werden, dass mit diesem System die derzeitigen Grenzen von programmierbaren Plattformen überschritten werden können. Verschiedene skalierbare Algorithmen wurden implementiert und im Gesamtsystem erfolgreich getestet.

Danksagung

Der Autor dankt allen Projektmitgliedern der Firma Philips in Eindhoven (Niederlande) und Briarcliff Manor (USA) für ihre wertvollen Beiträge und die Realisierung des Experimentiersystems. Ergänzend zu den in der Literatur [2, 4, 7, 8] aufgeführten Personen zählen I. Nitescu und C.M. Otero Pérez.

Literatur

- [1] E. Aarts, S. Marzano: The new everyday – Views on ambient intelligence. Koninklijke Philips Electronics N.V., OIO Publishers, Rotterdam, The Netherlands 2003.
- [2] C. Hentschel, R.J. Bril, M. Gabrani, L. Steffens, K. van Zon, S. van Loo: Scalable video algorithms and dynamic resource management for consumer terminals. International Conference on Media Futures (ICMF), Proc., Mai 2001, pp. 193-196.
- [3] R.J. Bril, M. Gabrani, C. Hentschel, S. van Loo, L. Steffens: QoS for consumer terminals and its support for product families. Int. Conference on Media Futures, Florence (Italy), Mai 2001, Proceedings, pp. 299-302.
- [4] S. Peng: Complexity scalable video decoding via IDCT data pruning. IEEE Int. Conf. on Consumer Electronics (ICCE), Digest of Technical Papers, Juni 2001, pp. 74-75.
- [5] R.J. Bril, L. Steffens, S. van Loo, M. Gabrani, C. Hentschel: Dynamic Behavior of Consumer Multimedia Terminals: System Aspects. IEEE Int. Conf. on Multimedia and Expo, Proceedings, Tokyo (Japan), August 2001, CD-ROM, ISBN 0-7695-1198-8.
- [6] R.J. Bril, C. Hentschel, E.F.M. Steffens, M. Gabrani, G. van Loo, J.H.A. Gelissen: Multimedia QoS in Consumer Terminals. IEEE Int. Workshop on Signal Processing Systems (SIPS 2001), Proceedings, Antwerpen (Belgien), September 2001, pp. 332-343.
- [7] T.-H. Lan, Y. Chen, Z. Zhong: MPEG-2 decoding complexity regulation for a media processor. IEEE Multimedia and Signal Processing Workshop (MMSP), October 2001, pp. 193-198.
- [8] C. Hentschel, R. Braspenning, M. Gabrani: Scalable Algorithms for Media Processing. Int. Conference on Image Processing (ICIP), Proc., October 2001, pp. 342-345.