Beherrschbare Zeit

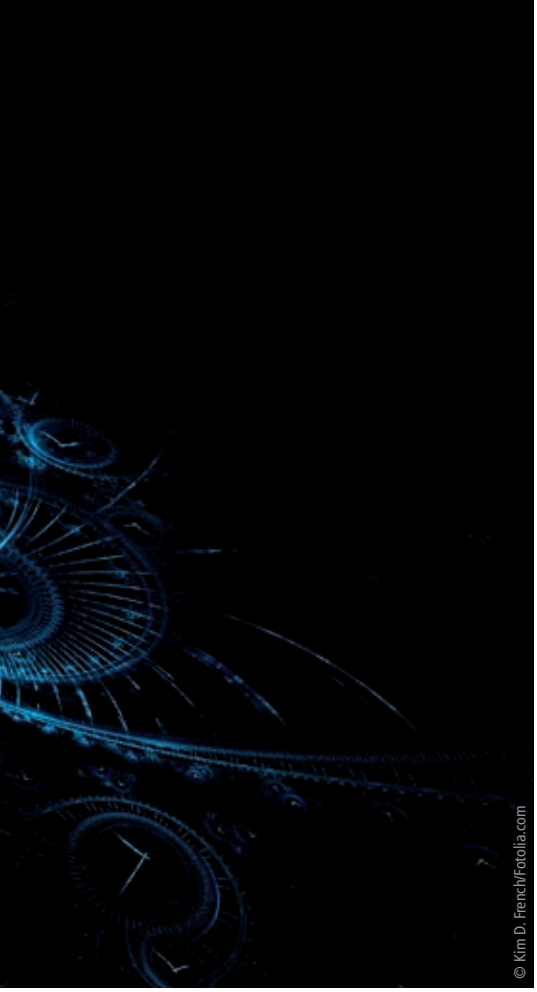
Grundlagen der Bildverarbeitung: Echtzeit

Der Begriff „Echtzeit“ wird in der industriellen Bildverarbeitung häufig benutzt, aber nur selten genau definiert. Oft wird die „Echtzeitfähigkeit“ eines Systems damit assoziiert, dass es „schnell“ ist, manchmal damit, dass es Bilder im Video-Takt auswerten kann. In diesem Beitrag wird eine Definition des Echtzeitbetriebs gegeben, die sich an den Anforderungen für Prüfsysteme in der laufenden Produktion orientiert. Die wesentlichen Aspekte sind dabei, dass ein Echtzeitsystem jederzeit auf asynchrone Ereignisse reagieren kann und innerhalb einer definierten Zeitspanne eine Rückmeldung gibt.

Interrupt

Bei Prüfaufgaben in der laufenden Produktion kommen die Teile meist nicht in einem festen Takt, sondern asynchron in den Sichtbereich der Prüfstation. Daher kann man sich den Zeitpunkt für die Bildaufnahme nicht frei aussuchen. Wenn ein Prüfobjekt vor der Prüfstation angeliefert wird, muss die Bildaufnahme innerhalb eines definierten Zeitfensters ausgelöst werden. Eine Prüfstation muss also ständig einen Sensor (z. B. eine Lichtschranke) auslesen bzw. auf ein externes Signal hin die Prüfung beginnen können. Sie muss dann die Bildaufnahme auslösen und möglicherweise auch andere externe Geräte ansteuern, z. B. eine Blitzbeleuchtung. Nach der Bildauswertung muss sie dann Informationen an den Prozess zurückgeben oder sogar selbst einen Aktor ansteuern, z. B. eine Weiche zum Aussteuern der Schlecht-Teile. Das Sensorsignal zur Bildaufnahme kann zu jedem beliebigen Zeitpunkt einlaufen. Ein solches asynchrones Steuersignal wird als „Interruptanforderung“ bezeichnet. Man sagt auch, das Prüfprogramm arbeite „ereignisgesteuert“, es muss zur „Ausnahmeverarbeitung“ in der Lage sein. Dahinter steht die Vorstellung, dass in der Prüfstation dauernd ein Programm abläuft, auch wenn kein Prüfling vor der Kamera ist. Es könnte z. B. sein, dass ständig Bilder aufgenommen werden oder eine Schnittstelle abgefragt wird. Wenn ein Prüfobjekt dann eine Lichtschranke abdeckt, wird „in Hardware“ die Spannung an einem Pin der Lichtschranke verändert, z. B. von „Low“ nach „High“ gezogen. Die Spannung dieses Pins wird auf einen Eingang des Bildverarbeitungssystems geschaltet und muss ständig vom System überwacht werden. Mikrocontroller haben eigene Register, die speziell zur Überwachung von Interrupts gedacht sind. Schon ein einfacher 8-bit-Controller hat mehrere Interrupt-Register zur Verfügung. Wenn ein Interrupt-Pin „hochgezogen“ wird, wird ein Flip-Flop gesetzt und der Interrupt gespeichert, so lange, bis er wieder gelöscht wird. Man prüft dann in regelmäßigen Zeitabständen nach, ob ein Interrupt-Flag gesetzt ist.

Wenn eine Interruptquelle im Betriebssystem eine ausreichend hohe Priorität hat, wird der normale Programmablauf unterbrochen und zu einer „Interrupt-Service-Routine“ verzweigt, sobald die Interruptanforderung vom System registriert wurde. Das kann man sich wie den Aufruf einer Funktion oder eines Unterprogramms vorstellen. Die Interrupt-Service-Routine übernimmt dann die vollständige Kontrolle über das System. Vorher muss der Status des übrigen Programms gesichert werden. Wenn die Interrupt-Service-Routine abgearbeitet ist, werden die Speicher so restauriert, dass der Zustand vor Einleitung der Interrupt-Verarbeitung wieder hergestellt ist, und das Programm läuft weiter. Der wesentliche Unterschied zwischen Interrupt-Service-Routine und einem Unterprogramm ist, dass das Programm an



einer beliebigen Stelle stehen kann, wenn der Interrupt auftritt. Ein Unterprogramm jedoch wird nur an genau definierten Stellen des Programmablaufs aufgerufen. Während der Abarbeitung der Interrupt-Routine kann grundsätzlich eine weitere Interruptanforderung auftreten, die wiederum den Programmablauf unterbrechen möchte.

Ausnahmeverarbeitung Bildaufnahme

Bei einer Prüfstation in der industriellen Bildverarbeitung werden die Bildaufnahme und die Bildauswertung also in der Interrupt-Service-Routine durchgeführt. Sie stellen demnach (in der Terminologie der Informatik) die „Ausnahmeverarbeitung“ dar. Vielleicht müssen Sie sich an diesen Gedanken erst einmal gewöhnen: normalerweise ist das Prüfprogramm damit beschäftigt, in einer Schleife auf die Ankunft eines Prüfobjekts zu warten. Dabei kann es natürlich allerlei nützliche Dinge tun, z.B. die Funktionsfähigkeit der Beleuchtung überprüfen. Dieser Betriebszustand wird durch die Ankunft eines Prüfobjekts unterbrochen, die Lichtschranke meldet sich, und die Ausnahmeverarbeitung – die eigentliche Prüfroutine – tritt in Aktion. Sie meldet anschließend das Ergebnis an eine übergeordnete Instanz, und das Prüfprogramm kehrt in den Normalzustand zurück. Wenn genügend viele Teile in genügend geringem Abstand aufeinander folgen, befindet sich das Programm überwiegend in der Ausnahmeverarbeitung. Genauso ist es aber möglich, dass nur alle zwei Sekunden ein Bild aufgenommen und ausgewertet werden muss, so dass sich das Programm (ausreichend schnelle Bildauswertung vorausgesetzt) praktisch ständig langweilt.

Das Programm einer Prüfstation in der laufenden Fertigung arbeitet also ereignisgesteuert: das Ereignis „Lichtschranke detektiert einen Prüfling“ löst die eigentliche Bildverarbeitungsroutine aus. Darüber hinaus kommt das Ereignis, wann es will, also asynchron zu den anderen Abläufen im Prozess. Das System muss also in jedem beliebigen Betriebszustand in der Lage sein, eine Interruptanforderung zu erkennen und die zugehörige Serviceroutine abzuarbeiten.

Schritthaltende Bildverarbeitung

Der Interrupt tritt auf, wann er will: das Programm kann gerade mitten in einer Funktion stecken oder am Beginn des Hauptprogramms. Außerdem können die Interruptquellen nur in einem festen Takt abgefragt werden, so dass von einem gesetzten Interrupt nur bekannt ist, dass er irgendwann zwischen der letzten und der vorletzten Abfrage gesetzt wurde, also innerhalb eines bestimmten Zeitfensters. Ob der Interrupt am Ende, am Anfang oder irgendwann während dieser Zeitspanne aufgetreten ist, kann nicht mehr festgestellt werden. Es verbleibt also eine zeitliche Unsicherheit. Das Interrupt-Management ist keine Trivialität. Während die Interrupt-Service-Routine läuft, kann unter Umständen durch die Lichtschranke ein weiterer Interrupt angefordert werden, je nachdem, wie die Information aus der Lichtschranke ausgewertet wird. Beispielsweise könnte bei einem ausgedehnten Objekt, das eine Lichtschranke unterbricht, der Interrupt dauernd wieder gesetzt werden. Möglicherweise ist es daher sinnvoll, den Interrupt erst dann abzusetzen, wenn das Objekt die Lichtschranke verlässt, wenn also die Lichtschranke von „unterbrochen“ auf „frei“ wechselt. Darüber hinaus ist es sinnvoll,

mit einem System zu arbeiten, das die auftretenden Interrupts speichern kann.

Die zeitliche Unsicherheit, mit der ein asynchroner Hardware-Interrupt festgestellt werden kann, führt zu einer Verschiebung der Position des Prüflings im Bild. Auch Sensoren und AD-Wandler haben Schaltzeiten. Hinzu kommt, dass die Bildaufnahme, in vielen Fällen auch eine Blitzbeleuchtung, bei bewegten Objekten in zeitlich definierten Abständen zum Interrupt ausgelöst werden müssen. Auch die Rückmeldung an den Prozess muss in dieser Hinsicht zeitlich definiert sein, damit Aktoren wirklich das richtige Objekt aussteuern oder greifen können. Die zeitlichen Verhältnisse in der Realität müssen also durch die Prüfroutine so genau abgebildet werden, dass alle mechanischen und elektrischen Komponenten des Systems die Prüfobjekte tatsächlich verfolgen und mit dem Fertigungstakt Schritt halten können. In diesem Zusammenhang sind nicht nur die Rechenleistung und die Taktfrequenz des Prozessors, sondern auch das Zeitverhalten der übrigen Hardwarekomponenten relevant.

Echtzeitfähigkeit

Ein echtzeitfähiges System muss in der Lage sein, in jedem beliebigen Betriebszustand eine (asynchrone) Interruptanforderung zu erkennen, den Programmablauf zu unterbrechen und die zugehörige Serviceroutine auszuführen. Allein dadurch wird aber die hinreichend genaue Abbildung des realen Zeitablaufs im System noch nicht gewährleistet. Es muss außerdem sichergestellt sein, dass das System nach einer definierten Maximalzeit nach Auslösen der Interruptquelle die Serviceroutine abgearbeitet und das Ergebnis an den Prozess zurückgemeldet hat, so dass der „Normalzustand“ wieder hergestellt ist. Man bezeichnet Systeme, für die eine solche maximale Reaktionszeit angegeben werden kann, als deterministisch oder vorhersagbar.

Die gesamte Reaktionszeit setzt sich aus folgenden Zeiten zusammen:

- Der Zeit, die erforderlich ist, um aus dem Signal an der Signalquelle (z. B. der Lichtschranke) eine Interruptanforderung zu erzeugen. Dabei gehen z. B. die Schaltzeiten von Gattern oder die Zugriffszeiten auf Interruptregister ein.
- Der Zeit, die das System benötigt, um die gesetzte Interruptanforderung zu erkennen. Dabei geht unter anderem die Zykluszeit des Prozessors ein.
- Der Zeit, die das System benötigt, um als Reaktion auf die erkannte Interruptanforderung die Serviceroutine aufzurufen. Bei vielen Betriebssystemen haben andere Vorgänge höhere Priorität als die Reaktion auf die Interruptanforderung.
- Der Zeit, die die Prüfroutine benötigt, um die Prüfaufgabe abzuarbeiten, inkl. Bildeinzug.

Die Summe der ersten drei Zeiten bezeichnet man meist als „Interrupt-Latenzzeit“. In dieser Zeitspanne ist die Interruptanforderung zwar im System vorhanden, muss aber auf ihre Realisierung warten. Diese Zeit wird in Datenblättern oder Publikationen meist angeben. Leider ist diese Zeit für ein gegebenes System oft nicht konstant. Man findet daher häufig Angaben zu „typischen Werten“, selten zu Maximalwerten, sehr selten zur Verteilung der Latenzzeiten. Erst aus der Verteilung kann man aber abschätzen, mit welcher Wahrscheinlichkeit höhere Latenzzeiten als die, die man im System eigentlich haben möchte, auftreten werden, und ob man das damit verbundene Risiko akzeptieren will. Nicht zu vergessen ist der vierte Zeitsummand: Mit der Programmierung der Prüfrou-

tine wird unmittelbar das Echtzeitverhalten des Systems beeinflusst. Eine Prüfroutine kann unterschiedliche Ausführungszeiten haben, je nach dem, wie die Bilddaten genau aussehen. Wenn z.B. bei einer Klassifizierung in einem Entscheidungsbaum nur selten in einen bestimmten Fall verzweigt wird, bei dem eine weitere Schleife abgearbeitet wird, kann die Ausführungszeit stark variieren. Deshalb ist es wichtig, die Verarbeitungszeit sorgfältig (d.h. für alle möglichen Zustände des Programms) zu testen oder wenigstens abzuschätzen, wenn die Echtzeitfähigkeit dadurch kompromittiert werden könnte. Wenn Programme so komplex werden, dass ein solcher systematischer Test nicht mehr möglich ist, kann die Echtzeitfähigkeit nicht mehr streng nachgewiesen werden. Kritisch sind in diesem Zusammenhang rekursive Verfahren, etwa bei der Bestimmung von Nullstellen oder bei Interpolationen, und die ständige Verfügbarkeit des erforderlichen Speichers.

Systemversagen

Ein Echtzeitsystem muss unter allen möglichen äußeren Bedingungen innerhalb eines vorgegebenen Zeitintervalls definiert reagieren – wenn diese Spanne überschritten wird, hat das System versagt. Insbesondere muss bei einem echtzeitfähigen Betriebssystem die Interruptquelle in der Regel die höchste Priorität im System erhalten, höher als jede Betriebssystemroutine. Viele weit verbreitete Betriebssysteme wie Windows oder Unix sind in diesem Sinne ohne Zusatzmaßnahmen nicht echtzeitfähig. Auch die Peripherie kann die Echtzeitfähigkeit eines Systems einschränken. Wenn z.B. eine Blitzlampe zur Beleuchtung verwendet wird, ist die minimale Zeit zwischen zwei Blitzen durch die Ladezeit des Blitzkondensators festgelegt. Das Betriebssystem kann dann in puncto Echtzeitfähigkeit beliebig gut sein – wenn die Wartezeit bis zum nächsten möglichen Blitz nicht eingehalten werden kann, wird kein verwertbares Bild des Objekts entstehen. Generell kommt es bei der Echtzeitfähigkeit darauf an, dass ein System ereignisgesteuert jederzeit ein Bild aufnehmen, das Bild auswerten und die Information so rechtzeitig an den Prozess zurückmelden kann, dass die erforderliche Aktion sicher ausgeführt werden kann. Wie lang die gesamte Reaktionszeit sein darf, damit die Bildverarbeitung mit dem Fertigungstakt Schritt halten kann, hängt von der Anwendung ab. Bei den typischen Anwendungen der industriellen Bildverarbeitung bilden Prüfaufgaben mit 100 Bildaufnahmen pro Sekunde und einer Transportgeschwindigkeit von 10 m/s glücklicherweise eher die obere Grenze der Anforderungen, so dass Reaktionszeiten im Bereich von einigen Millisekunden meist schon zu echtzeitfähigen Lösungen führen, solange die Vereinzelung der Teile funktioniert und ein spezifizierter Mindestabstand eingehalten wird. Im Vergleich zu den Echtzeit-Problemen bei der Realisierung von Airbag-Sensoren oder ABS und vielen anderen Anwendungen aus der Mess-, Steuer- und Regeltechnik sind wir damit in der industriellen Bildverarbeitung für die laufende Produktion mindestens mittelfristig in einem sicher beherrschbaren Bereich.

▶ **Autor**
Prof. Dr. Christoph Heckenkamp
Hochschule Darmstadt
Studiengang Optotechnik
und Bildverarbeitung
heckenkamp@h-da.de
www.fbmn.h-da.de