



Så fixar du syn på drönare och robot

Systemkrets och mjukvarustack stakar ut vägen

Drönare med kamera ombord är ett hett diskussionsämne just nu. Inbyggd avbildning kan delas upp i två kategorier: system som är medveten om sin omgivning samt system som både är medveten om omgivningen och som vidtar åtgärder.

Robotar och drönare som styrs med hjälp av bilder tillhör den senare kategorin. De reagera på olika situationer genom att först känna av, sedan bearbeta, analysera och fatta beslut, för att därefter kommunicera och styra.

Inom det civila används drönare exempelvis i medicinska tillämpningar, inom jordbruket, vid tv-sändning och filminspelning och för brottsbekämpning. I många tillämpningar erbjuder de stora fördelar. Inom exempelvis brottsbekämpning och filmproduktion kan de ersätta helikoptrar, och på så sätt minska kostnaden rejält.

Drönare kan också utföra mer skräddarsydda tjänster. Ett sådant exempel är den kommande leveranstjänsten Amazon Prime eller leverans av medicinska produkter till avlägsna områden, något som exempelvis sker i Uganda. Inom jordbruket kan drönare använda hyperspektral bildbehandling för att bestämma hur grödan mår.

Tillämpningarna som nämnts ovan är bara ett axblock av de som drönare redan används inom eller i framtiden kan komma att användas inom. Tillsammans demonstrerar de olika trender inom inbyggt seende:

- **För det första:** Intelligens genom distribuerad maskinlärning (at the edge). Inbyggd intelligens innebär att drönaren måste extrahera information från sina kameror och agera utefter den för att uppnå sitt mål.
- **För det andra:** Det krävs högnivåspråk och öppenkodsbibliotek för att implementera intelligens i en drönare. De vanligaste öppenkodsbiblioteken är OpenCV, OpenVX och Caffé. OpenCV är öppen källkod för datorseende, som innefattar ett antal bibliotek för datorseende i realtid baserat på C/C++. OpenVX är program med öppen källkod för utveckling av bildbehandlingsprogram för datorseende, medan Caffé är ett deeplearningbibliotek.
- **För det tredje:** Säkerhet i flera nivåer.



Av Giles Peckham, Xilinx och Adam Taylor, Adiuvo Engineering and Training

Giles Peckham är Marketing Director EMEA på Xilinx. Under sina år på företaget har han arbetat med både teknik och marknadsföring. Giles började inom halvledarindustrin för över 25 år sedan. Först som konstruktör av ASSP:er för konsumenttillämpningar på Philips Semiconductor (idag NXP), därefter med grindmatriser (gate array) och standardceller.



Adam Taylor äger konsultfirman Adiuvo Engineering and Training. Han är expert på utveckling och konstruktion av inbyggda system baserade på FPGA:er. Adam har skrivit ett stort antal artiklar om elektronik- och FPGA-konstruktion, däribland ett drygt hundratal blogginlägg om hur man använder Xilinx Zynq-kretsar.

Det krävs säkerhet på enhets-, system- och nätverksnivå för att säkerställa att drönare kan användas utan att säkerhet och information äventyras.

- **För det fjärde:** Tillgänglighet överallt. Än så länge finns inte robotar och drönare som styrs genom seende överallt, likt våra smartmobiler. Däremot ser tillämpningar som drar nytta av dem en stark tillväxt.

RENT KONSTRUKTIONSMÄSSIGT är en drönare uppdelad i tre kritiska delsystem: exakt motorstyrning i realtid, programvarudefinierad radio för dubbelriktad kommunikation samt mycket exakt seende. Till detta kommer att lösningen måste vara energieffektiv eftersom kamerastyrda drönare och robotar oftast är batteridrivna.

Ett avancerat inbyggt system för seende har hög bildhastighet och förmågan att bearbeta och agera på en bild snabbare än en människa. Många tillämpningar kräver stereo eller multipelt seende, då används flera kameror för att skapa ett komplett perspektiv av omgivningen runt drönaren. Kombinationen av flera sensorer i olika kombinationer för att skapa en fullständig bild av omgivningen kallas sensorfusion.

I många tillämpningar används som sagt flera kameror. I vissa fall används kameror som detekterar andra delar av det elektromagnetiska spektrumet, såsom i infraröda

och hyperspektrala tillämpningar. De kallad vanligen homogena om de använder flera sensorer av samma typ eller heterogena när de använder olika sensortekniker.

För att skapa förbättrad upplevelse och undvika hinder kör ett inbyggt visionsystem algoritmer som exempelvis Slam (Simultaneous localization and mapping) och snabbt optiskt flöde. Systemet kopplar dessa algoritmer med mer traditionella algoritmer för mönster- och objektigenkänning.

Kamerastyrda robotar och drönare kräver därmed inte bara hög bearbetningstakt med låg och deterministisk fördröjning. De måste också kunna optimeras med avseende på energiförbrukning och erbjuda en skalbar och framtidsbeständig lösning.

Det är här systemkretsarna Zynq-7000 och Zynq UltraScale+ kommer in. Det är två plattformar som kombinerar programmerbar logik med Arm-kärnorna A53 eller A9. För databehandling i realtid har Zynq UltraScale+ dubbla Arm Cortex-R5-processorer. Denna heterogena arkitektur bäddar för effektiv segmentering av funktionerna i antingen processorn eller den programmerbara logiken.

Med ett Arm-baserat system är det möjligt att utföra mycket sofistikerad bildanalys och beslutsfattande i realtid. I kombination med programmerbar logik kan systemkretsen upptäcka objekt i realtid



Inom jordbruket kan drönare med kamera användas för att se hur grödan mår och i staden krävs högt optiskt flöde för att undvika hinder.

och klassificera dem. När väl klassificeringen är klar kan roboten eller drönaren vidta lämpliga åtgärder för den detekterade objekttypen. Utvecklarna använder tekniker för maskininläring för att skapa förutsättning för distribuerade beslut i realtid.

Traditionellt har det krävts en expert på hårdvarubeskrivande språk som HDL för att implementera bildbehandling och maskininläring i programmerbar logik. Det har inneburit lång utvecklingstid och hög kostnad.

Mjukvarustacken reVISION kan användas med alla Zynq Soc och Zynq UltraScale+ MPSoC. Den gör att utvecklare kan arbeta direkt med standardramverk och -bibliotek för att skapa en systemmodell.

STACKEN reVISION är arrangerad i tre olika lager:

1) Plattformslagret. Det är den lägsta nivån i stacken och den som resterande lager är byggda på. Här ingår plattformdefinitioner av hård- och mjukvarumiljön. Om en utvecklare inte vill använda fördefinierade plattformar kan den istället skapa en skräddarsydd plattform med Vivado Design Suite.

2) Algoritm-layer. Konstruktionsmiljön SDSoc Design Environment och plattformdefinitioner av målhardvaran används för att skapa tillämpningen. Det är inom detta lager som utvecklare kan använda OpenCV-funktioner tillsammans med fördefinierade och optimerade implementeringar av konvolutionsnät (Convolutional Neural Network, CNN), såsom inferensacceleratorer. Detta går

att bygga effektivt i programmerbar logik.

3) Applikationslager. Det är det högsta nivån i stacken och där utvecklare använder ramverk som Caffe och OpenVX för att färdigställa tillämpningen.

MED REVISION kan utvecklare accelererar OpenCV-funktioner i den programmerbara logiken hos systemkretsarna Zynq Soc och Zynq UltraScale+. Det skapar ett mer responsivt bildbehandlingssystem, eftersom rörledning till sin natur är parallella. Den parallella bearbetningen i FPGA-strukturen tar bort den främsta flaskhalsen i GPU-konstruktioner, som skyfflar data fram och tillbaka mellan chip och DDR-minne. Likaså minskar effektförlusten när data hålls inom kretsen.

I tillämpningar med autonoma drönare och robotar används ofta maskininlärningsinferenser för att fatta beslut. Mjukvarustacken reVISION gör det möjligt att implementera inferensmotorer i den programmerbara logiken för att stödja detta.

För att hjälpa konstruktörer att implementera och träna neuronnät finns några ramverk som blivit branschstandarder, däribland Caffe. Integrationen med Caffe i reVISION är enkel. reVISION kan ta nätverks- och träningsinformation direkt från en Caffe-prototxtfil. Genom att använda programmerbar logik kan utvecklare accelerera en inferensmotor och därmed skapa en mer responsiv och energieffektiv lösning.

Talrepresentationen som används i inferensmotorn påverkar prestanda. Inom

maskininläring används oftast decimalkomma och reducerad precision, såsom åttabits heltal (Int8). Med hjälp av Int8 kan konstruktörer erhålla en maskininlärningsinferensmotor i programmerbar logik. Int8-operationen använder dedikerade DSP-block (DSP48E2) som finns i UltraScale+-arkitekturen.

DSP-blocken höjer prestanda då de är dedikerade MAC-block, utformade för fastpunktsaritmetik. Deras struktur bäddar också för resurseffektivitet, eftersom varje block kan utföra två Int8-MAC-operationer om de använder samma vikter. Metoden kan ge upp till 1,75 gånger högre genomströmning och två till sex gånger ökad energieffektivitet (uttryckt i giga-operationer per sekund per watt) jämfört med konkurrerande enheter.

reVISION minskar utvecklingstiden och ger en mer responsiv, energieffektiv och flexibel lösning vid konstruktion av bildbehandlingssystem för seende robotar och drönare. Jämfört med GPU-lösningar ger Zynq-7000 SoC eller Zynq UltraScale + MPSoC en femtedel av latensen. Likaså ökar prestanda och energieffektivitet motsvarande 6x (bilder per sekund per watt) vid maskininläringstillämpningar och upp till 42x (bildrutor per sekund per watt) vid så kallad embedded vision.

Visuellt styrda drönare och robotar måste också konstrueras med hänsyn tagen till säkerhet – allt för att hindra obehörig modifiering eller att någon kommer åt systemet och dess data. Även om säkerhet måste byggas från systemnivå och nedåt så erbjuder Zynq-7000 SoC och Zynq UltraScale + MPSoC flera säkerhetsaspekter på krets- och systemnivå. Utvecklare kan exempelvis kryptera och verifiera boot- och konfigurationsprocessen, samtidigt som de stödjer Trustzone-teknik. Med Trustzone går det att skapa ortogonala världar med hjälp av en hypervisor.

Likaså kan den inbyggda AD-omvandlaren eller systemmonitorn användas för att övervaka spänningar, temperaturer och externa parametrar för att på så sätt förhindra manipulering. Det finns även ett flertal andra konstruktionsval som ytterligare kan öka säkerheten – allt beroende av kravet på systemet. ■



Mjukvarustacken reVISION med tillhörande ramverk och bibliotek.

Level 1	Level 2	Level 3
Absolute difference	Channel combine	Box
Accumulate	Channel extract	Scale/Resize
Accumulate squared	Color convert	StereoRectify
Accumulate weighted	Convert bit depth	Median
Arithmetic addition	Convert bit depth	Warp Affine
Arithmetic subtraction	Histogram	Warp Perspective
Bitwise AND, OR, XOR, NOT	Grayscale Phase	Custom convolution
Pixel-wise multiplication	MinMax Location	Fast corner
Integral Image	Mean & Standard Deviation	Dilate
Gradient magnitude	Thresholding	Erode
		Inequalize Histogram
		Color Detection
		StereoLDM

OpenCV-funktioner för programmerbar logik.