

# MULTI-FOVEA ARCHITEKTÚRA ÉS ALGORITMUSOK CELLULÁRIS SOKMAGOS SZÁMÍTÁSI PLATFORMOKON

*Tézisfüzet a Ph.D. disszertációhoz*

**Soós Balázs Gergely**

[1][2][3] [4] [5] [6] [7]

Tudományos vezető:

**Rekeczky Csaba, Ph.D.**



Pázmány Péter Katolikus Egyetem  
Információs Technológiai Kar  
Multidiszciplináris Műszaki Tudományok Doktori Iskola

Budapest 2010

## 1. Bevezetés – Érzékelés és feldolgozás

Az Intel 1974-ben készített 8080-as processzorában 4500 tranzisztor volt. A chipgyártás technológiája robbanásszerű fejlődésen ment keresztül, így mára akár néhány milliárd tranzisztort is elhelyezhetnek egyetlen szilícium chipen [8]. Ez lehetővé teszi olyan chipek piaci megjelenését, amin sok ezer számítási egység (mag) található. Számítástudományi szempontból a jelen nagy kihívása, hogy milyen struktúrában kell ezeket elhelyezni, illetve milyen kommunikációs hálózatot kell kialakítani köztük, hogy a számítási kapacitást valóban ki lehessen használni: a magok ténylegesen számolhassanak és ne a következő bemeneti adatra kelljen várniuk, a kimenő adatok eljussanak a megfelelő helyre. Ki kell alakítani a struktúrákhoz illeszkedő algoritmusokat, melyek hatékonyan bontják a problémát párhuzamosan kiértékelhető szegmensekre. Az adat-kommunikációs idők optimalizálása kényszeríti ki a térbeli lokalitás precedenciáját, és celluláris számítási struktúrák felhasználását [9].

A félvezető technológia és egyéb anyagtudományi újítások másik fontos felhasználási területe az érzékelők gyártása. A tízmillió képpontot rögzítő digitális fényképezőgép mára tömegtermék. Léteznek kamerák, melyek több ezer képkockát rögzítenek másodpercenként, vagy éjszaka is látnak a fény infra tartományát érzékelve. Az orvosi diagnosztika ma már elképzelhetetlen képalkotó műszerek, röntgen vagy ultrahangos vizsgálat nélkül, de a fizika és számítástechnika találkozási újabb eszközöket is ad (mágneses rezonancia készülék – MRI, pozitron emissziós tomográf - PET).

Egy hétköznapi életben megszokott számítógépes probléma, szövegszerkesztés, vagy táblázatkezelés megoldásában a több magos processzorok nem sok gyorsítással kecsegtetnek, mert nem lehet a feladatot hatékonyan szétosztani. Az eddigi tendencia az volt, hogy a gyártók a fizikai korlátokat feszegetve egyre nagyobb frekvencián működő processzorokat kínáltak. A közelmúltban a multimédiás anyagok (fotók, videók, zene) széleskörű terjedésével új típusú adatokat kapott a személyi számítógép. Ezek nagyfokú strukturáltsága lehetővé teszi, hogy megjelenítés, szerkesztés során a képeket (zenét), részekre bontsuk, és szétosszuk a feldolgozó egységek között, a részeredmények összefésülésével pedig előáll a kívánt eredmény. Ez a piaci szegmens komoly fejlesztéseket indukált a többmagos számítási platformok területén. Tudományos számítások, különböző fizikai, kémiai, biológiai folyamatok modellezésére a problémák mérete miatt a párhuzamosítás igénye a kezdetektől jelen van. Számos fontos jelenség dinamikus rendszerek kölcsönhatásával írható le [10], melyek számítógépes modellezéséhez szükséges az egészezen értelmezett algoritmus-fogalom kiterjesztése. A problémák celluláris jellege természetes módon vezet el topologikus struktúrába rendezett processzortömbökig, a celluláris hullám-számítógépekhez [11][12].

Az érzékszervek által nagy mennyiségben begyűjtött adathalmazban megtalálni a lényeges információt, és az alapján dönteni, kulcsfontosságú valamennyi élőlény számára. A biológiai látó rendszerekben nagyfokú konvergencia figyelhető meg az érzékelő idegsejtek és az agykérgi feldolgozás között. A szenzorokhoz kapcsolódó idegsejtek hálózata nagymértékben párhuzamos előfeldolgozást végez, így válik lehetővé az,

hogy a több millió receptor által összegyűjtött információt néhány százezer idegsejt továbbítsa az agy felé. A retinára vetődő látvány idegsejtes reprezentációja nem egyenletes. Jóval sűrűbb receptor- és feldolgozó hálózat tartozik egy kitüntetett központi elhelyezkedésű területhez, az úgynevezett foveához. A fovea a szemgolyó elmozdulásával a látvány más és más részleteire irányítható.

Az élő rendszerek vitathatatlan hatékonysága ötletet ad a mesterséges látó-rendszereket tervező mérnök számára: párhuzamos strukturált feldolgozás, egy-egy kiemelt terület részletes vizsgálatával. Ha az előfeldolgozás során több régió is fontosnak tűnik, mindegyiket elemezhetjük. Ez a multi-foveális koncepció. A sikeres mérnöki megoldások pl. [13] megmutatták, hogy a modell valóban felhasználható, megteremtve az igényt egy egységes algoritmikus keretrendszer elkészítésére.

A mozgó kamerás távfelügyelet egy fontos és nehéz problémakör. Például kisméretű robotrepülő (Unmanned Aerial Vehicle - UAV) felhasználása nagy kiterjedésű területek megfigyelését teszi lehetővé: a biztonságtechnikai, közlekedés-szervezési adatgyűjtés mellett komoly anyagi károkat enyhíthet katasztrófa-helyzetekben egy átfogó légi felvétel. Ahhoz, hogy a lehető legtöbb hasznos információt gyűjtse be a rendszer, szükség lehet a mozgó platformon lejátszódó döntésre, mely módosítja az előre tervezett repülési pályát az érzékelt látvány alapján. A repülő platform véges energia-ellátása és a gyors beavatkozás szükségessége egy hatékony vizuális navigációs rendszer kialakítását igényli.

## **2. Kutatási célkitűzés, eredmények**

A munkám során strukturált adatokon, elsősorban képfolyamokon definiált problémák hatékony párhuzamos megoldásának elősegítését tűztem ki célul, celluláris számítási struktúrák és a multi-foveális megközelítés együttes felhasználásával. Elméleti szempontból fontos megoldandó kérdés volt egy egységes algoritmus-környezet kialakítása, mely hatékonyan elősegíti a párhuzamos hardver megválasztását, konkrét eszköz elfedését. Gyakorlati szempontból kapcsolódik a feladathoz a vizuális navigációs rendszereknél felhasználható algoritmusok elemzése.

A kidolgozott platform specifikus párhuzamos tömbprocesszorokat tartalmaz az előfeldolgozás és a foveális feldolgozás támogatására. A javasolt heterogén struktúra jól illeszkedik a nagyfokú strukturáltsággal rendelkező bementő videófolyamhoz. Tervezési utmutatásokat adtam a Multi-Fovea Architektúra hatékony felhasználásához, melyeket mini robotrepülő vizuális navigációs berendezésében futtatható algoritmusok elemzésével együtt mutattam be. Az algoritmusok földön mozgó objektumok detektálását teszik lehetővé 2D regisztrációs megközelítéssel. Az algoritmusok analitikus összehasonlítása után javasoltam egy új megoldást, mely az algoritmus nagyobb százalékát képes párhuzamosan futtatni a beépített celluláris sturktúrák kihasználásával.

### 3. Vizsgálati módszerek

A kutatói munka elindulásakor az emberi látórendszerrel kapcsolatos neurobiológiai kutatások és más neuromorf modellek motiváltak [14][15].

A kidolgozott architektúrális modell elméleti háttérét főként a geometria és a képfeldolgozás, a párhuzamos számításelmélet, a topologikus celluláris operátorok és algoritmusok [16], valamint a gráfelmélet területéről összegyűjtött eredmények adják.

A célorientált, lokális kommunikációt preferáló (celluláris), virtuális és fizikai architektúrák tervezése [12] lényegi eleme a módszernek.

Az algoritmusok leírására használt struktúra matematikai értelemben irányított aciklikus gráf (DAG), ami ütemezési feladatok absztrakt megoldására széleskörben használt. A leírás a tisztán celluláris (CNN) algoritmusok leírására használt UMF diagram [17] kiterjesztésének tekinthető.

A munkám során az asztali számítógépen (PC) futó szimulációs környezetet készítettem, emellett számos tömbprocesszor implementáción (ACE16K [18], EyeRis [19], Nvidia Geforce8800 [20]) végeztem kísérleteket, a rendszerekhez tartozó specifikus szoftverek és programozási nyelvek felhasználásával.

Az ALFA projekt [9] kapcsán bekapcsolódtam a mérés-adatgyűjtési folyamatba, így a 3D szimulációs környezetben előállított mesterséges video mellett valós, Gödöllői repülőtér felett készített légi-felvételeken is tesztelhettem az algoritmusokat. Saját felvételeket is készítettem siklóernyőről.

A módszerek összehasonlító analízisére, egységes futtatására Matlab, illetve Matlab Simulink [9][21] rendszereket használtam, egyes modulokat C++ nyelven implementáltam, illetve felhasználtam referencia implementációkat.

## 4. Új tudományos eredmények

*I. A topografikus érzékelők jelfolyamait feldolgozó algoritmusok az adat természetes strukturáltsága miatt nagyfokú adatpárhuzamosíthatóságot tartalmaznak. Erre a megfigyelésre alapozva kidolgoztam egy virtuális hardver architektúra modellt (Multi-fovea architektúra), mely támogatja közvetlen szenzoros adatot feldolgozó, konvergens, aciklikus adatfolyamgráffal leírható képfeldolgozó algoritmusok kommunikáció-hatékony dekompozícióját fizikai hardver architektúrára. A javasolt struktúra változó topológiájú és lefedettségű, különböző mértékű csatolást igénylő operátorokat tartalmazó algoritmusok hatékony futtatását teszi lehetővé három, a műveletek sajátosságaihoz illeszkedő processzortömb összekapcsolásával. Ez a hibrid struktúra az uralkodó homogén felépítésű többmagos architektúráknál a feladatosztályra jobban illeszkedő, heterogén programozható eszközt eredményez.*

**Kapcsolódó publikációk:** [1][5]

Az egészek, vagy lebegőpontos számok feldolgozására épített processzorok egyszerre egy operandus halmazon (általában két szám) végeznek egy műveletet előállítva az eredményt. A processzorok általában néhány szám tárolására alkalmas kitüntetett belső tárolót, regisztert tartalmaznak, melyeken közvetlenül lehetséges a műveletvégzés. A további adatok egy külső memória-egységben vannak tárolva. A programok adatmozgató és matematikai utasítások



sorozatából állnak. A kiértékelés valamely adat függvényében elágazhat (branching), vagy ismétlődhet (ciklusok).

A feldolgozás gyakran az adathalmaz szomszédos elemeit érinti. Ez a tulajdonság a lokalitás. A lokalitás a programkódban is megfigyelhető, az elágazások ritkán fordulnak elő, az utasítások egymás után rendezhetők a memóriában. A feldolgozott adat mennyisége miatt a processzor mellett külön memória-chipet szoktak a számítógépekbe építeni. Ez azt jelenti, hogy a memória-hozzáférés chipen kívüli kommunikációt igényel. Ez technológiától függően 2-3 nagyságrenddel nagyobb késleltetésű, mint a processzor számítása, ezért érdemes az adat és a program lokalitását kihasználva a külső memóriákból nagyobb részt lemásolni a belső úgynevezett cache-memóriába.

A felhasználói programok túlnyomó többsége dominánsan soros probléma, egy végrehajtási szállal jól leírható. A program által előállított részeredmények általában nem rögtön kerülnek felhasználásra, illetve több részeredményt kapcsol össze a következő bináris művelet. Ilyenkor a két résszámítás sorrendje felcserélhető, illetve akár párhuzamosan is végrehajtható (utasítás szintű párhuzamosság). Az utasítások végrehajtása, és a memóriaműveletek legtöbbször átlapolhatók. Az optimalizáláshoz szükség van az adatok és az utasítások előre olvasására. A modern PC-kben lévő processzorok több párhuzamos számítási sort tartalmaznak, illetve adatra várakozás esetén futtatás közben újrasorrendezik az egymástól nem függő utasításokat (out-of-order execution). Az elágazások kezelésénél szükség van mind a két lehetséges ág kezelésére, vagy valamilyen heurisztika alapján az egyik kiválasztására. A strukturált adatok megjelenése előtérbe hozta néhány

elemű vektorokat egy lépésben kezelő vektor-utasítások és hozzá tartozó nagyméretű regiszterek megjelenését, tovább növelve a processzorok komplexitását.

Nagyméretű problémák megoldására egyetlen processzor gyakran nem elég gyors. Ebben az esetben meg kell találni az algoritmus párhuzamosan megoldható részeit, és szét kell osztani a feladatot az eszközök között. Az egyes processzorokat külön-külön el kell látni programmal és adattal, meg kell oldani a kommunikációt a részeredmények cseréjére, illetve a lépések szinkronizációjára. Ez a fajta programozási módszer új kihívásokat jelent: biztosítani kell az eszközök hozzáférését a közös erőforrásokhoz (legkritikusabb a memória és a kommunikációs hálózat), a cachelt memóriaterületek koherenciáját. A processzorok soros végrehajtási sebessége fizikai korlátok miatt már nehezen növelhető, de több számítási mag elhelyezése egy chipen megoldható, ezért a nehézségek ellenére is a párhuzamos kiértékelés kerül előtérbe.

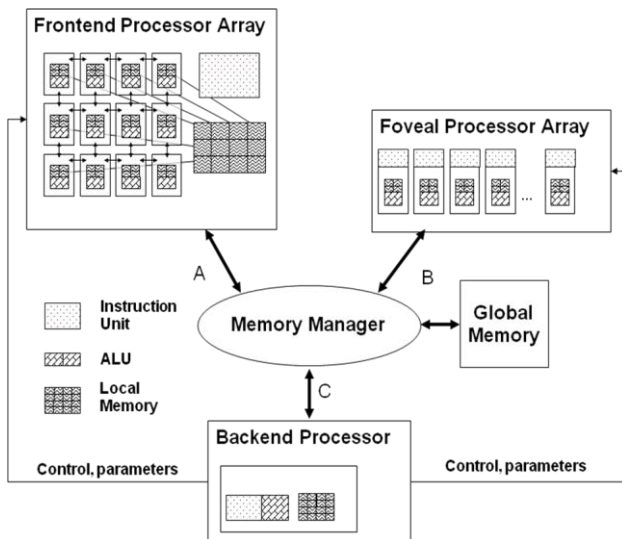
Imperatív programozási nyelven egy nagy adathalmazt elemenként ciklussal tudunk bejárni. A multimédiás adatok nagyfokú adatpárhuzamosságot tartalmaznak. Valamennyi elemet azonos módon kell feldolgozni legtöbbször csak a szomszédos adatoktól függően. Az ilyen típusú műveletek a celluláris operátorok. Ez lehetőséget nyújt a ciklus párhuzamos végrehajtására: több végrehajtási szál indítható. Az adott operátor ismert lokalitása miatt a nagyméretű automatikus cache lecserélhető az egyes szálakhoz rendelt, programozottan feltöltött lokális memóriára. Az legelterjedtebb imperatív nyelv a C. A szálak kezelésére egy kiterjesztése jött létre, az OpenMP [22], mely a lokalitási viszonyok

explicit megadása után a program fordítása során a ciklusokat párhuzamosítja többmagos processzoron történő végrehajtásra.

A párhuzamos adatszerkezeteken leírt operátorok a konkrét bejárési és vezérlési struktúra explicit megkötése nélkül, csak a lokalitási viszonyok megadásával még nagyobb fordítás-időbeli optimalizálást tesznek lehetővé (RapidMind [23], Intel Threading Building Blocks [24]). Az operátorok függőségi viszonyait egy soros megadás esetén fel kell térképezni. A adatfolyamgráf-szerű megadás a programozó számára is kényelmes reprezentáció, emellett explicit módon tartalmazza a fogyasztó-termelő kényszereket.

Párhuzamos problémák megoldására cél-processzorok hatékonysága vitathatatlan, hiszen egy konkrét algoritmus struktúrájához tökéletesen illeszkednek, de az átprogramozhatóság feladása mellett. Homogén sokmagos struktúrák alkalmazása általános célú és jobban skálázható számítási platform lehet, de egyes algoritmus osztályok támogatására hatékonyabb hibrid architektúrák is kialakíthatók.

Egy virtuális heterogén sokprocesszoros architektúrát dolgoztam ki ( 1. ábra ), közvetlen szenoros adatot feldolgozó, konvergens, aciklikus jelfolyamgráffal leírható képfeldolgozó algoritmusok megvalósítására. A virtuális architektúra lehetővé teszi a fizikai megvalósítás elfedését szoftvertervezési szempontból, a kommunikációs és időzítési viszonyok figyelembe tartása mellett.



1. ábra A Multi-fovea architektúra funkcionális elemei:

Celluláris topologikus processzortömb (Frontend Processor Array)

Foveális ablakozó processzortömb (Foveal Processor Array)

Soros processzor (Backend Processor)

Az egyes processzortömbök sok számítási egységet tartalmaznak (ALU), melyeket utasítás dekódoló végrehajtó egységek (Instruction Unit) irányítanak. A tömbök tartalmaznak gyors elérésű lokális memóriát (Local Memory), a nagy méretű, de lassú elérésű rendszer memória (Global Memory) mellett. Az ablakozást, és a memóriák elérését az intelligens memória menedzser egység (Memory Manager) teszi lehetővé az A,B,C kommunikációs vonalakon keresztül. Az egyes tömbök szinkronizálását, és programmal való ellátását, a részeredmények végső kiértékelését a soros processzor (Backend Processor) végzi.

Az általam javasolt virtuális architektúra a cellulárisan lokális adatpárhuzamosság kihasználására egy dedikált celluláris topologikus processzortömböt tartalmaz (Frontend Processor Array). Ez az egység a teljes bemenő kép lokális operátorokkal ([25]) felírt előprocesszálására alkalmas. Egyetlen utasítás-végrehajtó egységet tartalmaz, így elsősorban elágazás-mentesen kiértékelhető, tér- és adat-invariáns műveletekre hatékony.

A képfeldolgozó algoritmusok többségének célja a képen valamely régió megjelölése (szegmentáció, klasszifikáció), objektum vagy esemény felismerése (detekció, azonosítás), vagyis a topologikus műveletek után bizonyos képrészletekre jobban fókuszál. Az előprocesszálás során megtalált régiók részletes kiértékelését a foveális processzortömb segíti (Foveal Processor Array). Az egyes ablakok feldolgozása elágazásokat tartalmazhat adat és vezérlési szinten is, ezért a tömb aritmetikai egységeit (vagy azok csoportját) különálló utasítás-végrehajtó egység irányítja. Az egyes foveák között szükséges kommunikáció és szinkronizáció közvetlenül valósul meg. A foveális futások eredménye lehet ablak méretű képrészlet, vagy valamilyen leíróvektor.

A nem párhuzamosítható programrészek, a leíróvektorok feldolgozását a soros processzoron történik (Backend Processor). Ez az egység felelős a többi egység szinkronizálásáért is. Az adatmozgatást (teljes képek, ablakok, skalárok) egy intelligens memóriavezérlő egység támogatja.

Az általam javasolt virtuális architektúra alkalmas a megcélzott problémaosztály esetén a megvalósító hardver hatékony elfedésére. Az egyes képszintű operátorokhoz és kommunikációs műveletekhez végrehajtási idő rendelhető, a különböző algoritmusok összehasonlíthatóvá válnak. A hardveres megvalósítás szempontjából legfontosabb, hogy az operátorok lokalitás viszonyai, és topológiája explicit módon megadásra kerül. Elkülönülnek egymástól az elágazást tartalmazó és nem tartalmazó vezérlési szakaszok. Tervezhető az

adatmozgatás ütemezése a nagyobb memóriablokkok együttesen kezelhetők.

Az egyik meghatározó PC-s környezetbe beépülő programozható, párhuzamos architektúra az Nvidia cég videokártyáin megtalálható nagyszámú (akár néhány száz) általános célú grafikus magra (GP-GPU) épülő "Egységesített hardver Architektúra" (Compute Unified Device Architecture-CUDA) [22]. Az AMD-ATI videokártyák is GP-GPU-kat tartalmaznak, de a hozzájuk kapcsolódó programozói környezet a grafikai alkalmazásokhoz egyelőre erősebben kötődik.

A CUDA modell és a Multi-fovea architektúra analógiáit és eltéréseit vizsgáltam. Az ablakozó processzortömb és a soros processzor mindkét modellnek eleme. A multi-fovea struktúra egyediségét az adja, hogy tartalmaz egy dedikált, sűrű celluláris összekötöttségű, 2D topologikus feldolgozó processzortömböt is. Az előfeldolgozási lépések során a képfeldolgozó algoritmusok nagyon gyakran az egész képet érintő topologikus műveleteket végeznek. A műveletek ezen nagy családja (pl. konvolúció, morfológia) csak kis sugarú lokális csatolást tartalmaz, de ez minden pixelt egyenletesen fed le, így a kiértékeléshez particionálás esetén nagymértékű, de lokális kommunikáció szükséges. Megmutattam, hogy a GP-GPU architektúrán is hatékonyan megvalósítható a Frontend Processor Array, tehát a virtuális architektúra ennek a platformnak a lefedésére is alkalmas .

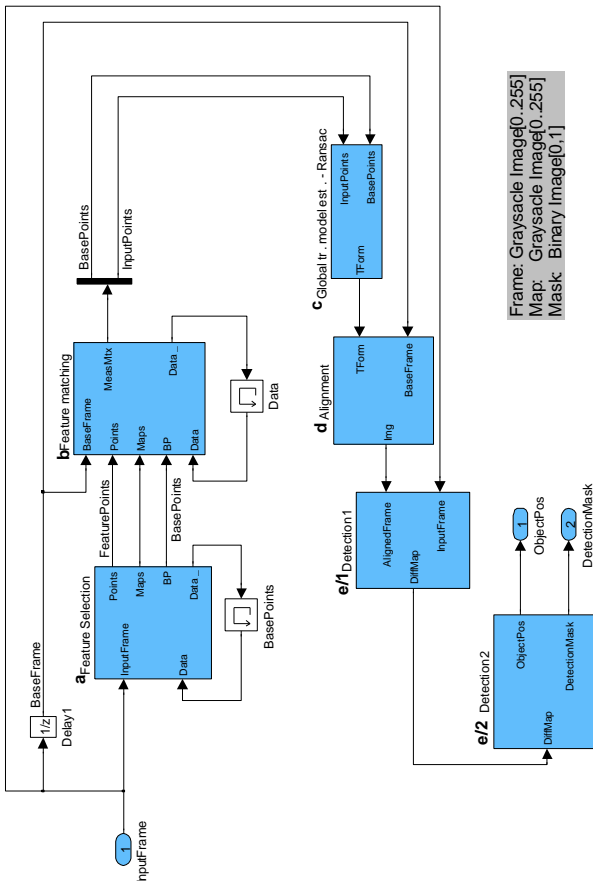
*II. Megmutattam, hogy az általam javasolt virtuális architektúra alkalmas képfeldolgozó, képkalkoló algoritmusok széles körének egységes leírására. Konkrét példaként mini robotrepülő (UAV) látórendszerében a földön mozgó objektumok detekciójára alkalmazható 2D regisztrációs algoritmusokat mutattam be. Elvégeztem a területen legtöbbet hivatkozott algoritmusok leírását az adatfolyamgráf megadásával, az operátoraik processzortömb hozzárendelését, és módszerek egységes szimulációs keretben történő összehasonlítását. Meghatároztam a fenti algoritmusok számítási komplexitás, regisztrációs minőség, detekciós robusztusság és paraméter érzékenység szerinti kritikai elemzését. A részletes elemzés után komplexitás csökkentő módosításokat javasoltam, melyeket két altézisben fejtek ki.*

**Kapcsolódó publikációk:** [4]

Nagy kiterjedésű sík megfigyelési terület felett 80-100 méteres magasságban cirkáló robotrepülők felvételein 2D-s kép-regisztrációs technikákkal azonosíthatók a földön mozgó járművek [26]. A sík terület különböző pozíciókból rögzített vetületei közös koordináta rendszerbe transzformálhatók. A talajon mozgó objektumok regisztrációs eltérést eredményeznek.

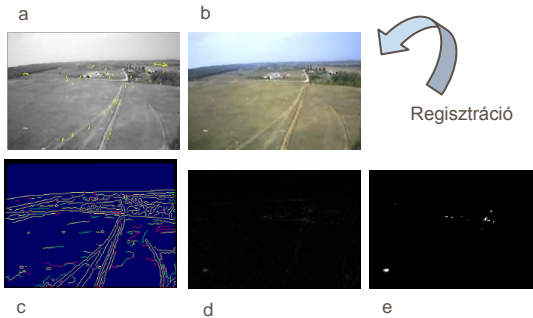
A kulcsponthoz (valamilyen szempontból kiugró, robusztusan azonosítható képpont, pl. élek találkozásánál létrejövő sarok) alapú összerendelés az egyik meghatározó algoritmus család a 2D

regisztrációs megoldások között. Ezek a módszerek természetes módon felírhatók adat-parallel formában.



2. ábra **Globális regisztráció alapú detekciós algoritmusok általános folyamatábrája**  
 Az aktuális és az előző képkockán detektált kulcspontok (a) párokba rendezése (b) után a globális transzformációs mátrix kerül meghatározásra (c). Az előző képkocka összes képpontja leképezhető az új képkocka koordináta-rendszerébe (d), és kiszámítható a változást leíró szűrkeármlyalatos detekciós kép (e/1), melyből megfelelő binarizálás után származtatható a detekciós eredmény (e/2).





**3. ábra Globális regisztráció alapú detekciós algoritmusok részeredményei:**

az összerendelt kulcsponatok az előző képkockán ábrázolva (a); az aktuális bemenet (b); a szűrkeárnyalatos detekciós kép (d). A mozgó objektumok fehér foltot eredményeznek a bináris detekciós képen (e), a regisztráció minőségét az egymásra vetített élképek mutatják (c).

A szakirodalomban kiemelten tárgyalt kulcspon kereső és regisztrációs megoldások, a Harris féle sarok detektor (Harris Corner Detector, Corner Pairing - CPA) [27], és a Kanade-Lucas kulcsponkövető (Kanade-Lucas Tracker – KLT) [28][29], mellett a legtöbbet hivatkozott megoldás a skála invariáns leíró transzformáció (Scale Invariant Feature Transform – SIFT) [30]. Felhasználhatók továbbá a video tömörítés miatt folyamatosan optimalizált blokk-párosítás (block-matching algorithm - BMA) metódusok [31].

A fenti módszerekre épített robusztus regisztrációs [32] és detekciós algoritmusokat írtam le multi-fovea megközelítésben. Több elemzés is foglalkozik a fenti módszerek minőségének összehasonlításával [33] [34], de hatékony implementálhatóságuk összehasonlító elemzéséről nincs tudomásom. Az algoritmusok általános folyamatábrája a 2. ábrán található, a 3. ábra a részeredményeket szemlélteti.

Több valós és egy szimulált videofolyam felhasználásával összehasonlítottam az algoritmusok komplexitását, felhasználhatóságukat mozgó megfigyelőtől független mozgás meghatározásához. Szokásos terület megfigyelés során a repülő négyzetes pályát jár be, hosszú egyenes szakaszok után kezd forduló manőverbe. A jelenetekben az egyenes szakaszokról lettek kiválasztva. A repülőgép magassága a szakaszok mentén folyamatosan változhat. A szimulált videó képe éles, előre kiszámolt pontokból pillanatfelvételek lettek kiszámolva. A valós felvételeken a mozgás miatt fellépő elmosódás jelen van. A felvételek 320x240 pixeles felbontásban készültek, 20 képkocka per másodperc sebességgel. Az egyes képkockák közti eltolás maximális mértéke 12 pixel volt. A nagyobb mintavételi frekvencia segítségével az algoritmusok jelentősen egyszerűsíthetőek. Megfelelően érzékeny szenzor mellett ideális lenne 1-2 pixels tartományba szorítani az elmozdulást.

Az összes algoritmus a kulcspontok összerendelése után lényegében azonos regisztrációs és detekciós modult használt.

A CPA, BMA, KLT algoritmusok komplexitását döntően két változtatható paraméter befolyásolja: a foveális ablakok száma és a párosításnál felhasznált képrészlet mérete. Az előfeldolgozó rész komplexitása érdemben nem paraméterezhető. A SIFT algoritmusnál az előfeldolgozó rész komplexitása jelentősen paraméterezhető.

Az algoritmusok detekciós robusztusságát a kiválasztott paraméterek megkötése mellett vizsgáltam. Az 1. táblázat eredményei szerint a módszerek hasonlóan teljesítenek, jelentősen eltérő komplexitás mellett.

Amennyiben a feladat túlmege a detekción és pontos szegmentáció, vagy azonosítás a cél a KLT és SIFT algoritmusok jóval nagyobb komplexitása megtérülhet. A pontosabb regisztrációnak köszönhetően.

A regisztráció jóságát a nagyfrekvenciás tartományban értékeltem ki, a bináris élképek átfedésének arányával (EEdge) jellemezve. Pontos regisztráció esetén a képkockánkénti transzformációk akumulálhatók, hosszabb kép-szekvenciák rendezhetők azonos koordináta rendszerbe, így lassan mozgó objektumok is detektálhatóvá válnak.

Az algoritmusok különböző hardver komponensek mellett tekinthetők ideálisnak. A SIFT minőségét tekintve kiemelkedő módszer, az objektumok azonosítására is alkalmas leíróvektort állít elő. Hátránya, hogy nagyon erős előfeldolgozó, és ablakozó processzortömböket igényel. KLT módszer fovea intenzív számításokat igényel, törtek ábrázolásával, és nem topologikus műveletek kiértékelésével, de egyszerűbb előfeldolgozást, mint a SIFT algoritmus. Regisztrációs minőségük egyformán kiemelkedő.

A BMA módszerek előfeldolgozó tömb nélkül működnek. A különböző stratégiák: kimerítő-keresés illetve gyémánt-keresés jó kompromisszumot kínálnak a komplexitás és pontosság között. A kimerítő-keresés számszerűen nagy számításigényű, de mivel elágazásmentes a programja, így egyszerűbb fizikai tömb is képes futtatni.

A CPA algoritmus regisztrációs minősége szerényebb, de a detekciós feladat megoldására alkalmas, kis előfeldolgozási és foveális igény mellett.

Az elemzések közben összegyűjtött tapasztalatok alapján komplexitás csökkentő módszereket határoztam meg, melyeket altézisekben fejtek ki.

**II.1. Megmutattam, hogy közepes magasságban repülő UAV felvételeinek kiértékelésekor skálatérben történő kulcspont kereséshez nincs szükség felskálázásra (az eredeti szenzorfelbontásnál nagyobb rácson történő számításra). Kísérletileg igazoltam továbbá, hogy a video- folyamatban kis képkockák közti nagyítás (<1%) várható, így kettőnél több oktávon belüli skálafelosztásra nincs szükség.**

A SIFT algoritmus térbeli Gauss szűrők (aluláteresztő, a képet simító szűrő) sorozatával, és a szűrt képek különbségének kiszámításával a bementeti kép olyan transzformáltjait hozza létre, melyeken a foltszerű képrészletek robusztusan lokalizálhatók. A szűrő  $\sigma=2$  paraméter melletti futtatásával az effektív képfelbontás felére eszik, illetve a folyamatos skála térben 1 oktávval. Lowe javaslata szerint [30] ahhoz, hogy optimális detekciót érjünk el, szükség van a kép dupla méretű reprezentációjára, és oktávonként három további finom felosztásra.

A dedikált 2D celluláris processzortömbön hatékonyan kiszámítható a Gauss szűrőknek megfelelő diffúzió [18], de a tömb méretének a szenzorfelbontás kétszeresére történő emelése legalább négyszeres hardverkomplexitást igényelne, míg az oktávonkénti finom felbontások lineáris növekedést jelentenek.

A regisztrációs minőség elemzése során megmutattam, hogy megfelelő kulcspár összerendelés érhető el két oktávonkénti beosztás esetén is, illetve azt, hogy elegendő kulcspár keletkezik a felskálázás

nélkül is a robusztus transzformáció-becslés végrehajtásához. Az egyszerűsített algoritmus az eredeti paraméterekkel futtatott változattól detekciós robusztusságban átlagosan nem tér el.

**II.2. Megmutattam, hogy a 2D celluláris processzáló tömbön hardveresen implementált diffúzió és 3x3-as környezetben működő extrémum kereső operátorok esetén a KLT és a SIFT algoritmusok ideálisan kombinálhatók.**

A skálatérben történő detekcióhoz szükséges 3x3x3 pixeles környezetben működő extrémum-hely kereső operátor. Ez programozottan megoldható már létező processzortömbön, de direkt hardveres implementálásának sincs akadálya. Az előző altézisben leírt egyszerűsítések után a kulcspontr detekció celluláris tömbprocesszoros implementációjának futásideje milliszekundumos tartományba kerül.

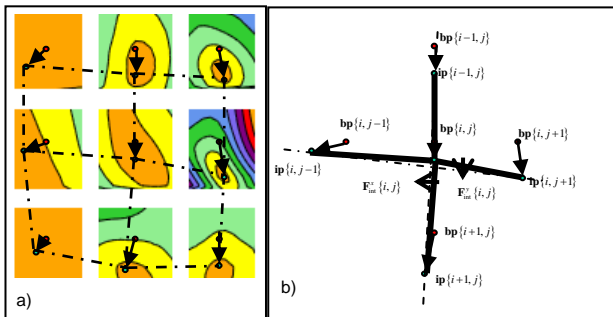
A további fontosabb megfigyelések a következők (1) a forgatás invariáns lokális mozgás-modell kiértékelése nem jár jelentős javulással, (2) a KLT algoritmus sub-pixel pontosságú regisztrációs megoldása kevés keresőablak esetén is a regisztrációhoz elegendő pontpárt állít elő. A tulajdonságok azt jelzik, hogy a kombinált skálatér-alpú kulcspontr kinyrés és a KLT szerű pont összerendelés hatékony megoldást kínál.

***III. Egy új független mozgás detekciós algoritmust dolgoztam ki (Elasztikus-háló multi-fovea detektor, Elastic Grid Multi-Fovea Detector - EGMD), mely hatékonyan kihasználja a javasolt hardver architektúra párhuzamos és lokálitási lehetőségeit. Kísérletileg igazoltam, hogy a felhasznált sokrégiós elmozdulás modell alkalmas földön mozgó objektumok megtalálására. Komplex minőség metrika térben pedig megmutattam, hogy a módszer hatékonyabb azonos környezeti feltevéseket használó korábbi megoldásoknál.***

**Kapcsolódó publikációk:** [2] [3]

Az előzőekben megvizsgált regisztrációs módszerek globális 2D elmozdulás modellt feltételeznek az egymás utáni képkockák regisztrációjakor (projektív transzformáció), ami közvetlenül illeszkedik a sík környezeti feltevéshez. Ennek meghatározása komplex lebegőpontos műveleteket, végrehajtása nem folytonos memória hozzáférési-mintát igényel. A javasolt párhuzamos, celluláris processzortömbökhöz illeszkedő felügyelt lokális mozgásmodell alkalmazása lehetővé teszi kisebb domborzati egyenetlenség tolerálását.

A módszer elve a 4. ábrán látható. A számítás dominánsan a foveális processzortömbön történik. A lokális részeredmények iteratívan konvergálnak, a szomszédos foveák aktuális eredményei alapján.



#### 4. ábra Elasztikus-háló multi-fovea detektor algoritmus

A kulcspontok hozzárendelése kereséssel történik, foveális feldolgozással. Egy fix rács csúcspontjaihoz (bázis pontok, bp) rendel az algoritmus megfelelő párt. Az egyes foveák az előző képből kivett kis képrészlethez leginkább hasonlító darabot keresik az aktuális képkockán. Az (a) kép az illeszkedési térképet mutatja, a rács csúcspontjaihoz viszonyított lehetséges lokális eltolások függvényében. A vektorok végpontjai (ip) feszítik ki az Elasztikus-Hálót (b). A háló iteratív optimalizáláson keresztül jut el a végső alakjára. A képrészletek hasonlóság-mértéke, és az élek találkozási meredeksége alapján vezérli a backend processzor a foveális processzortömb egyes egységeit. A szürkeárnyalatos és bináris detekciós képek meghatározása is foveálisan történik, a kiszámolt eltolási vektorok alapján.

	Artificial	Godollo_1	Godollo_2	Godollo_3	Komplexitás (művelet/pixel)
	135/130	120/79	300/230	35/31	
SIFT	130	52	200	29	~1100
KLT	130	52	217	29	~450
BMA	128	61	208	29	~100
CPA	125	55	183	27	~75
ELG	92	52	194	28	~50

#### 1. táblázat Érvényes-pozitív detekciós eredmények, és számítási komplexitás a vizsgált algoritmusoknál

A táblázat fejlécében szerepel az egyes szekvenciák hossza, illetve az, hogy az '1'-es célpont hány képkockán szerepel ('/'-jellel elválasztva). A táblázat sorai az egyes szekvenciákhoz tartozó referencia bejelölésekhez viszonyított érvényes-pozitív detekciós számát tartalmazzák, az egy pixelre vetített átlagos műveletszám megjelölésével.

Komplex összehasonlító mérésekkel bemutattam a módszer hatékonyságát egy szimulált és több valós felvétel kiértékelve. Az Elasztikus-háló multi-fovea detektor algoritmus szokásos terület-felügyeleti megfigyeléshez tartozó repülési pálya mellett átlagosan hasonló detekciós eredményeket ad, mint a korábbi algoritmusok, jelentősen kisebb számítási igény mellett (1. táblázat).

## 5. Az eredmények felhasználási területe

A Multi-fovea architektúra szenzorhoz kötődő kisméretű, kompakt beágyazott detektáló-klasszifikáló rendszerek alapvető számítástechnikai modellje lehet a közeljövőben. A VisCube projekt keretében egy multi-foveális architektúrát megvalósító chip tervezése és kísérleti gyártása folyik. Az algoritmikus vizsgálataim nagyban elősegítették az előprocesszáló réteg hardveresen megvalósított utasításkészletének meghatározását.

Multi-fovea architektúra emulálható más sokmagos rendszerben is (pl. FPGA-k). A hozzá kapcsolódó, és a disszertációban kifejtett tervezési módszerek segíthetnek komplex video feldolgozó algoritmusok széles körének párhuzamos processzormagokon történő implementálásában. Sok neves gyártó (pl. IBM, Intel, Nokia, Apple) támogatja az OpenCL nevű szabványt, ami koncepciójában nagyon hasonlít a CUDA-ra. A platformok várható elterjedése széleskörű lehetőséget biztosít az eredményeim felhasználására.



## 6. Köszönetnyilvánítás

Mindenekelőtt köszönettel tartozom témavezetőmnek, Rekeczky Csabának, aki tudományos pályára csábított, és töretlen bizalommal támogatta munkám az elmúlt években, bátorított az akadályok leküzdésében. Lehetővé tette számomra, hogy több alkalommal Berkeleybe látogassak, nemzetközi projektekhez kapcsolódhassak.

Köszönöm Roska Tamásnak, a PPKE ITK Multidiszciplináris Műszaki Tudományok Doktori Iskolájának és a SZTAKI Celluláris Érzékelő és Hullámszámítógépek Kutatólaboratórium vezetőjének, hogy lehetőséget biztosított számomra az itt folyó *Ph.D.* tanulmányaimra, és hogy türelmes mentorom volt az évek során.

A doktori munkám elvégzéséhez szükséges anyagi és szellemi feltételeket valamint az infrastruktúrát a Pázmány Péter Katolikus Egyetem Információs Technológiai Kara (PPKE ITK) és a Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutatóintézete (MTA SZTAKI) biztosította.

Külföldön töltött tanulmányútjaim során szakmai vezetőim voltak Ronald Tetzlaff Frankfurtban, a Johann Wolfgang *Goethe Universität-en*, és Wolfgang Porod Notre Dame-ben (USA, Indiana), a University of Notre Dame-en.

Köszönöm a SZTAKI laborban közvetlen munkatársaimnak, Zarándy Ákosnak, Földesy Péternek, Kék Lászlónak, Orzó Lászlónak, Szatmári Istvánnak, Radványi Andrásnak, és Szolgay Péternek, hogy megmutatták a mérnöki kutatásban rejlő szépséget, az alapokban rejlő érdekességet.

A disszertáció alapjául szolgáló munkában témavezetömön kívül közvetlen segítséget nyújtottak külföldi és hazai kollégáim: Cserey György, Rák Ádám, Veres József, Szabó Vilmos, Szálka Zsolt, Hillier Dániel, Gunter Geis, Sándor Alpár.

A Pázmány Egyetemen folyó oktatási munkába Vágó Zsuzsa és Bércesné Novák Ágnes segítségével kapcsolódhattam be, megismerve a fiatalokat. Roska Tamással és Rekeczky Csabával együtt dolgozva találtam meg Szabó Vilmost és Gelencsér Andrást, akik diplomázó diákjaim voltak.

Köszönöm Cserey Györgynek hogy összefogta többünk gondolatait, és kitarító támogatásával segített, hogy eredményekké váljanak.

Közvetlen munkatársaim ötleteire mindig számíthattam a felmerülő technikai problémák leküzdésében az ITK robotika laboratóriumába lépve.

Köszönöm munkatárain támogatását jelenlegi munkahelyemen, a SearchLab Kft-nél. Hornák Zoltán és Kiss Balázs értékeis hozzászólásai hozzájárultak az irásmű tökéletesedéséhez.

Itthoni és külföldi doktorandusz évfolyamtársaim, kollégáim szakmai és baráti támogatása nélkülözhetetlen volt, köszönöm mindannyiuknak: Bankó Éva, Benedek Csaba, Hegyi Barnabás, Lombai Ferenc, Harczos Tamás, Szálka Zsolt, Gyimesi Gergely, Weiss Béla, Zeffler Tamás, Ercsey-Ravasz Mária, Giovanni Pazienza, Bérci Norbert, Feldhoffer Gergely, Tar Ákos, Vásárhelyi Gábor, Iván Kristóf, Karacs Kristóf, Kiss András, Frank Gollas és Martin Eichler.

Köszönöm a támogatást Notre Dame-i patrónusaimnak Csurgay Ildikónak és Csurgay Árpádnak, és társaknak: Szakmány Gergő, Varga Edit, Tahy Kristóf.

Köszönöm az Analogic Kft, és az Eutecus Inc. munkatársainak, különösen Horváth Istvánnak, Gyimesi Tibornak és Erdősi Gábornak, hogy szakmai tanácsokkal segítettek a hardveres problémáim leküzdésében.

Csókási Anna, Adorján Livia, Tihanyi Judit, Vida Katinka közreműködése nélkülözhetetlen volt az adminisztratív feladatok elvégzésében. Köszönöm az egyetem gazdasági és adminisztratív dolgozóinak segítségét.

A kutatómunkához további anyagi támogatást nyújtottak magyarországi és USA K+F projektek: Országos Tudományos Kutatási Alapprogramok (OTKA), NI61101 számú program; a Nemzeti Fejlesztési Terv Gazdasági Versenyképesség Operatív Programja (GVOP), 3.1.1-2004-05-0388/3.0; Nemzeti Kutatási és Fejlesztési Programok, ALFA-2/046/04, TELESENSE-035/02/2001; az Office of Naval Research (ONR, USA), Multidisciplinary University Research Initiative (MURI) programja és az Eutecus Inc. VisCube 07-09 programja (ONR).

Köszönet ezen kívül az PPKE-ITK, és MTA-SZTAKI intézményekben dolgozó valamennyi kollégámnak.

Családom és Zsófi szeretete, bátorítása végig kísért a doktoranduszi éveim során, ezért soha nem apadó hálával tartozom.

## 7. Publikációk

### Folyóiratban, könyvfejezet

- [1] Soós, B. G., A. Rák, J. Veres, and G. Cserey, “GPU boosted CNN simulator library for graphical flow based programmability,” *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 2009, p. 11.
- [2] Soós, B. G., V. Szabó, and C. Rekeczky, “Elastic Grid Based Multi-Fovea Algorithm for Real-Time Object-Motion Detection in Airborne Surveillance,” *C. Bhaatar, W. Porod and T. Roska : Cellular Nanoscale Sensory Wave Computing*, Springer, 2009.

### Konferenciákon, Technikai Riport

- [3] B.G. Soós and C. Rekeczky, “Elastic Grid Based Analysis of Motion Field for Object-Motion Detection in Airborne Video Flows,” *Circuits and Systems, ISCAS 2007. IEEE International Symposium on*, 2007, pp. 617-620.
- [4] B.G. Soós, V. Szabó, and C. Rekeczky, *Multi-Fovea Architecture and Algorithms for Real-Time Object-Motion Detection in Airborne Surveillance*, Budapest, Hungary: Pazmány Peter Catholic University, 2009.
- [5] B.G. Soós, A. Rák, J. Veres, and G. Cserey, “GPU powered CNN simulator (SIMCNN) with graphical flow based programmability,” *Cellular Neural Networks and Their Applications, CNNA 2008. 11th International Workshop on*, 2008, pp. 163-168.
- [6] A. Rák, F. Gergely, B.G. Soós, and G. Cserey, “CPU-GPU hybrid compiling for general purpose: Case studies,” *Cellular Neural Networks and Their Applications, CNNA 2010. 12th International Workshop on*, 2010.

### Társszerzőként

- [7] A. Rák, B.G. Soós, and G. Cserey, “Stochastic Bitstream Based CNN and its Implementation of FPGA,” *International Journal Circuit Theory and Applications*, vol. published online, Nov. 2008.

## 8. A témához kapcsolódó irodalom

- [8] B. Stackhouse, S. Bhimji, C. Bostak, D. Bradley, B. Cherkauer, J. Desai, E. Francom, M. Gowan, P. Gronowski, D. Krueger, C. Morganti, and S. Troyer, “A 65 nm 2-Billion Transistor Quad-Core Itanium Processor,” *Solid-State Circuits, IEEE Journal of*, vol. 44, 2009, pp. 18-31.
- [9] ITRS, “International Technology Roadmap for Semiconductors,” <http://www.itrs.net>, 2009.
- [10] A. Turing, “The chemical basis of morphogenesis,” *Philosophical Transactions of the Royal Society (part B)*, vol. 237, 1953, pp. 37-72.
- [11] L.O. Chua and T. Roska, *Cellular neural networks and visual computing: foundation and applications*, Cambridge University Press, 2002.
- [12] T. Roska, “Cellular Wave Computing in Nanoscale via Million Processors,” *C. Bhaatar, W. Porod and T. Roska : Cellular Nanoscale Sensory Wave Computing*, Springer, 2009.
- [13] C. Rekeczky, I. Szatmari, D. Balya, G. Timar, and A. Zarandy, “Cellular multiadaptive analogic architecture: a computational framework for UAV applications,” *Circuits and Systems I: Regular Papers, IEEE Transactions on [Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on]*, vol. 51, 2004, pp. 864-884.
- [14] B. Roska and F. Werblin, “Vertical interactions across ten parallel, stacked representations in the mammalian retina,” *Nature*, vol. 410, 2001, pp. 583-587.
- [15] D. Bálya, B. Roska, T. Roska, and F.S. Werblin, “A CNN framework for modeling parallel processing in a mammalian retina,” *International Journal of Circuit Theory and Applications*, vol. 30, 2002, pp. 363-393.
- [16] L. Kek, K. Karacs, and T. Roska, *Cellular Wave Computing Library V2.1*, Budapest: Cellular Sensory Wave Computers Laboratory, Computer and Automation Research Institute, Hungarian Academy of Science, 2007.

- [17] T. Roska, "Computational and computer complexity of analogic cellular wave computers," *Cellular Neural Networks and Their Applications, 2002. (CNNA 2002). Proceedings of the 2002 7th IEEE International Workshop on*, 2002, pp. 323-338.
- [18] A. Rodriguez-Vazquez, G. Linan-Cembrano, L. Carranza, E. Roca-Moreno, R. Carmona-Galan, F. Jimenez-Garrido, R. Dominguez-Castro, and S. Meana, "ACE16k the third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, 2004, pp. 851-863.
- [19] AnaFocus, "eye-RIS 1.1 Leaflet of AnaFocus Ltd," 2007.
- [20] NVIDIA, *GeForce 8800 GPU Architecture Overview*, NVIDIA Corp., 2006.
- [21] MathWorks, "Simulink - Simulation and Model-Based Design," <http://www.mathworks.com/products/simulink/>, 2008.
- [22] R. Chandra, R. Menon, L. Dagum, D. Kohr, D. Maydan, and J. McDonald, *Parallel Programming in OpenMP*, Morgan Kaufman, 2000.
- [23] Rapidmind, "RAPIDMIND," <http://www.rapidmind.net/>.
- [24] "Intel Performance Libraries," 2008.
- [25] A. Zarándy and C. Rekeczky, "Many-core Processing Array Architecture Selection Guide for Topologic Problems," *C. Bhaatar, W. Porod and T. Roska : Cellular Nanoscale Sensory Wave Computing*, Springer, 2009.
- [26] R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, Hai Tao, Yanlin Guo, K. Hanna, A. Pope, R. Wildes, D.A.-H. Hirvonen, M.A.-H. Hansen, and P.A.-B. Burt, "Aerial video surveillance and exploitation," *Proceedings of the IEEE*, vol. 89, 2001, pp. 1518-1539.
- [27] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings Fourth Alvey Vision Conference*, Manchester, UK: 1988, pp. 147--151.
- [28] B.D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *International Joint Conference on Artificial Intelligence*, Vancouver: 1981, pp. 674-679.
- [29] J. Shi and C. Tomasi, "Good features to track," *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, 1994, pp. 593-600.
- [30] D.G. Lowe, "Distinctive Image Features from Scale-Invariant

- Keypoints,” *International Journal of Computer Vision*, vol. 60, Nov. 2004, pp. 91-110.
- [31] S. Zhu and K. Ma, “A new diamond search algorithm for fast block-matching motion estimation,” *Image Processing, IEEE Transactions on*, vol. 9, 2000, pp. 287-290.
- [32] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2004.
- [33] K. Mikolajczyk and C. Schmid, “A Performance Evaluation of Local Descriptors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, 2005, pp. 1615-1630.
- [34] B. Zitova and J. Flusser, “Image registration methods: a survey,” *Image and Vision Computing*, vol. 21, Oct. 2003, pp. 977-1000.