Gépi látás

II. rész

Cikkünk előző részében áttekintettük a gépi látás problémakörét, megismerkedtünk több konkrét alkalmazási területtel, végül a megoldás tipikus lépéseit vettük számba. Bár – remélhetőleg – érdekes témákkal, problémákkal foglalkoztunk, az átlagos videós életét ezen eredmények nem nagyon könnyítik meg. Éppen ezért az alábbiakban azokat a módszereket, eljárásokat igyekszünk sorra venni, melyek a „gépi látás laborokban" születtek, és nagyban segítik, vagy segíthetik olvasóink munkáját (hobbiját).


© ATR

Objektumok követése

A filmek és videók digitális feldolgozása során számtalan szűrőt, speciális effektust alkalmazhatunk a képeken: módosíthatjuk a színeket, a világosságot, a kontrasztot, elmoshatjuk vagy élesíthetjük a képet vagy éppen fénysugarakat generálhatunk. De gondoljunk például a családi videofelvételekre, ahol minden stimmel, csak éppen a Balaton színe borzalmas. Ebben az esetben nem az egész képet, hanem csupán egy bizonyos területet akarunk módosítani. Semmi probléma, hiszen már a legegyszerűbb vágóprogramok is képesek szín vagy világosság alapján szegmentálni a képet. Ha ez mégsem vezetne eredményre, a legtöbb esetben lehetőség van „megrajzolni" a maszkot, amit egy képrészlet elkülönítésére használhatunk.

De mit tegyünk, ha a kamera nem állványon állt, hanem kézből készítettük a felvételeket? Hogyan tartsuk rajta a kézzel rajzolt maszkot a mozgó háttéren? Ha megpróbáljuk kockáról kockára animációs kulcsokat létrehozva kiigazítani a maszk helyzetét, jó eséllyel hosszú órákig kell dolgoznunk egy viszonylag rövidebb anyagon is. Mint arról korábban már szóltunk, mozgó objektumok követése videofelvételeken az egyik legrégebben kutatott probléma, és mára már szerencsére szinte az összes vágó, kompozitáló és egyéb effektszoftverben megtalálható az ún. „tracker". Ez az eszköz képes a mozgókép egy – a felhasználó által meghatározott – darabkáját kockáról kockára követni.


Pontok követésével (tracking) a felhasználó által rajzolt maszk követi a mozgó objektumot

Az algoritmus működése az előző számban olvasható lépésekkel történik: minden újabb képkockán megjósoljuk a rendelkezésre álló információk (= a követett pont mozgása az előző képeken) alapján, hogy hozzávetőlegesen merre lesz a követendő pont. Ezután különböző matematikai eljárásokkal megkeressük azt az immár pontos pozíciót, amely a lehető legjobban megfelel a követelményeknek. Jelen esetben a követelmény: a felhasználó által meghatározott referencia képrészlet – például a főszereplő orra – a lehető legjobban hasonlítson a képsorozat éppen feldolgozott egy darabjára. A legjobb illeszkedés határozza meg az új kulcspozíciót.

Ha a videó egy pontját követjük le, az a maszk pozíciójának a vezérlésére alkalmas. Fejlettebb programokban lehetőség nyílik több pont követése által az objektum irányának és méretének követésére is. Az objektumkövetés videós szoftverekben jelenleg elérhető legfejlettebb változata a teljesen automatikus alakzatkövetés. Ebben az esetben a felhasználónak nem is kell jól azonosítható, és végig látható pontokat keresnie, ugyanis az algoritmus teljesen önállóan keres pontokat a megadott maszkon belül, és robusztusan – azaz az esetleges hibás pontok kiszűrésével – meghatározza az objektum mozgását.

Optical flow

Telhetetlenebb és/vagy leleményesebb olvasóinkban nyilván felmerült a gondolat: miért bízzák a programok a felhasználóra a követendő pontok kiválasztását? Miért nem követik a videó összes képpontjának mozgását? Így – többek között – az is teljesen

Az optical flow: megadja képről képre a képpontok mozgását

egyértelmű lenne, hogy a képen látható bármely objektum hogyan mozog. Természetesen ez a technika létezik, sőt napjainkban egyre több program képes az ún. optical flow számítására.

Egy képsorozatban minden egyes képpont követése igen nehéz feladat. A két legfontosabb probléma a számítás komplexitása (azaz a többmillió képpont feldolgozása), valamint az, hogy a legtöbbször nem is lehet egyértelműen megmondani, hogy mi a jó megoldás. Ez utóbbi első hallásra furcsának tűnhet, de sajnos, ami az embernek egyértelmű, az a számítógép számára gyakran nem az. És még véletlenül se gondoljuk, hogy ez a számítógépek, vagy az algoritmusok hibája lenne! Mint arról már az előző számban is írtunk, az emberi látás alapvetően nem analízisen, hanem felismerésen alapszik. Ha – akár fél szemmel is – látunk egy fehér falat, valamint azon egy sötétebb árnyékot, akkor a fényviszonyok alapján rögtön meg tudjuk állapítani, hogy egy mélyedés vagy egy kitüremkedés látható. De honnan tudjuk, hogy nem ügyes kezek festették az árnyékot a falra, és igazából egy tökéletesen sík felületről van szó? Sőt, ha nem is tudjuk, mik a fényviszonyok, könnyen összekeverjük a mélyedést a kitüremkedéssel! Az emberi érzékelést pont amiatt lehet viszonylag könnyen becsapni, hogy a hiányos információkat agyunk – tény, hogy igen jó hatásfokkal – kiegészíti. A számítógépes algoritmusok erre sokkal kevésbé képesek.

Ha a képsorozat egy teljesen homogén, azaz egyszínű részét próbáljuk meg lekövetni, az algoritmus nem képes megmondani, hogy milyen irányban mozognak a pixelek. Nem sokkal könnyebb a helyzet, ha egy egyenes vonalat kell követni: nem lehet eldönteni, hogy a vonal „hosszában" mozog-e, és ha igen, mennyit. Az ilyen eldönthetetlen helyzetek feloldására az optical flow algoritmusok bizonyos előfeltételekkel élnek, melyek – bár nem mindig állják meg a helyüket – lehetővé teszik a megoldást, ami természetesen nem biztos, hogy tökéletes lesz. Az egyik legfontosabb ilyen feltételezés az, hogy a szomszédos pixelek mozgása hasonló. Ez alapvetően nem is hangzik rosszul, de bizonyos esetekben sajnos nem igaz: például objektumok határvonalánál és tükröződő vagy átlátszó felületeknél.

Az optical flow gyakorlatilag egy irányokat tartalmazó képet jelent, ahol minden pixelhez hozzárendelünk egy 2 dimenziós mozgásvektort. Ha a kép pixeleit ezen vektorok mentén elmozgatjuk, akkor – elvileg – pontosan a következő képkockát kapjuk. Ezen vektorok kiszámítása a legtöbbször nem úgy zajlik, hogy minden egyes képpontot – egymástól függetlenül – lekövetünk a korábban megismert tracking algoritmusok egyikével, hiszen ez egyrészt kezelhetetlenül lassú lenne, másrészről a homogén képrészekkel nem is tudnánk mit kezdeni. A legtöbb eljárás egy sokkal hatékonyabb megközelítést alkalmaz: először a videó minden egyes képét ún. kép-piramissá alakítja, azaz eltárolja a felére, negyedére, nyolcadára stb. kicsinyített képeket. Ezután két szomszédos kép közötti optical flow kiszámítását a „legkisebb szinten" kezdi, ahol az icipici képméret miatt egyrészt nagyon kevés képpontot kell feldolgozni, másrészt feltételezhető, hogy adott képpontok mozgása igen kicsi. Ennek megfelelően viszonylag gyorsan le lehet követni az összes pixelt ezen az alacsony felbontású verzión. Ezután lépésenként mindig az eggyel jobb felbontású képpárokat vesszük elő, és finomítjuk az előző lépésben megkapott eredményt. Mindezt folytatjuk addig, amíg meg nem kapjuk az eredeti felbontású optical flow-t.

Eredeti képkocka

A mozgásvektorok segítségével a két kép közé generált új képkocka. Ezzel a technikával utólag lelassíthatjuk a felvételeinket

Eredeti képkocka

Ha ismerjük az összes képpont mozgását, azt igen sokféle speciális hatás létrehozásához felhasználhatjuk. Az egyik leggyakoribb ilyen átalakítás a film lelassítása (retime), ahol a rendelkezésre álló képkockák „közé" újakat generálunk. Ha a film sebességét az 50%-ra szeretnénk változtatni, nem kell mást tenni, mint a mozgásvektorok segítségével eltolni a pixeleket „félútra", így létrehozva a két szomszédos képkocka közötti állapotot. Szintén fel lehet használni az optical flow-t az utólagos „bemozdulás", azaz motion blur létrehozásához. Ha hangsúlyozni szeretnénk, hogy egy tárgy gyorsan

mozog a képen, akkor egyszerűen elmossuk a képet a mozgásvektoroknak megfelelő irányba. Ha nincsenek a képen mozgó tárgyak – például egy utca látható a videón – akkor a képpontok mozgása, és az objektumok távolsága gyakran összefügg. Így akár az is lehetségessé válhat, hogy utólag „belepiszkáljunk" a kép mélységélességébe!

Fontos megjegyeznünk, hogy a pixelek mozgásának elemzése a modern videotömörítő kodekeknek szintén alapvető része. Itt nem a tökéletes pontosság, hanem a nagy sebesség a fő kritérium, ezért a hierarchikus megközelítés helyett az ún. block matching algoritmust szokták használni. Ez azt jelenti, hogy felosztjuk a képet blokkokra, és ezeket a nagyobb területeket követjük le. Bár az így kapott mozgásvektorok nem lennének alkalmasak például a videó lelassítására, a tömörítést még így is nagyban segítik.

Restoration

Az előzőekben megismert technika segítségével meg tudjuk mondani, hogy a képen látható objektum – legyen az a háttér vagy egy szereplő – pontosan hol volt és hogyan nézett ki az előző és a következő képkockákon. Ez az ismeret tette lehetővé, hogy két szomszédos (n és n+1) kép közé köztes képkockákat generáljunk, a felvétel sebességének lassításához. Ugyanezt az eljárást alkalmazva az (n)-edik és (n+2)-ik kocka analízisével kiszámíthatjuk a köztes, (n+1)-ik képkockát. De miért akarnánk olyan képet létrehozni, ami eredetileg is a rendelkezésünkre állt? Természetesen azért, mert a celluloid filmre forgatott filmek bizony elég sok koszt, piszkot vagy egyéb sérülést tartalmazhatnak, így a képek kisebb-nagyobb darabjait utólag kell rekonstruálni.

Mára egész iparág jött létre a filmszalagok digitális restaurációjára, ahol aktívan alkalmazzák a gépi látás vonatkozó eredményeit. Meglepő módon nem csak a régi, jobbára fekete-fehér filmek szorulnak digitális „kozmetikázásra", hanem szinte minden film, amit celluloid szalagra rögzítettek. Ezen filmek rögzítésénél, laborálásánál, digitalizálásánál, bár csak kis mértékben, de elkerülhetetlenül sérülnek, koszolódnak a szalagok. A viszonylag jó minőségű anyagok tisztogatását „dust busting"-nak szokták hívni, a digitális filmrestaurálás inkább a rossz minőségű filmek feljavítását jelenti. A következőkben megismerkedünk néhány tipikus problémával, és röviden utalunk is a megoldás módjára.

Az egyik legnehezebb feladat a nagy kiterjedésű sérülések (pl. szakadás vagy nagy folt a kép közepén), vagy komplett (hiányzó) képkockák helyreállítása. Ebben az esetben szinte megkerülhetetlen az imént említett optical flow technika, mellyel sikeresen tudunk új képeket létrehozni az ismert, környező képek alapján. Ennek alkalmazása ott válik nehézzé, ahol a mozgásvektorok kiszámítása amúgy is problémákba ütközne: a filmen látható objektum számottevően változik (torzul), hirtelen új dolgok jelennek meg vagy tűnnek el, átlátszó felületeknél, illetve objektumok határánál, a kontúrvonalaknál. Sajnos ezek – különösen az utolsó példa – viszonylag gyakran előfordulnak, de az okozott hiba általában nem szembetűnő, hiszen jobbára csupán egy-egy képkockát módosítunk.

Kicsit könnyebben megoldható, de sokkal gyakoribb probléma a függőleges karcok kijavítása. Ezeket jellegzetes alakjuk és mozgásuk alapján viszonylag könnyű detektálni, a karcok vékonysága miatt a javítás is egyszerűbb. Itt is szoktak mozgásvektorokat számolni, de a gyakorlatban az egyszerűbb eszközök néha – gyakran(?) – hatékonyabbak. Mivel a karcok általában nem csak 1-2 képkockán keresztül, hanem akár másodpercekig is látszanak, nem jutunk sokkal előbbre, ha az egyes képeket a szomszédok segítségével akarjuk javítani. Ehelyett gyakran kielégítő eredményt nyújt az ún. benövesztés. Ez azt jelenti, hogy a karc két szélén látható – korrekt – pixelek színével kiszínezzük a sérülést. Mivel a karcok függőleges irányban futnak, itt a benövesztést oldalról végezzük, azaz egy átmenetet képezünk a sérülés egyik oldalától a másikig.

Sérült filmkocka

Digitális feldolgozással kijavítható a hiba

Az apróbb koszok, porszemek, kicsi foltok javítása önmagában nem jelent nagy problémát (használhatjuk a mozgásvektorokat vagy a benövesztést), de az automatikus detekció bizony nem egyszerű! Nagyobb digitális filmlaborokban előfordul, hogy szakmunkások hada tölti azzal az idejét, hogy a szkennelt filmen kockáról kockára, apró koszokat jelölgetnek be. Az automatizálás azért nehéz, mert a koszok könnyen összetéveszthetőek a filmen látható apró részletekkel, textúrák elemeivel. Az utóbbi években szerencsére nagy előrelépés történt ezen a területen, ugyanis a legmodernebb filmszkennerek már képesek a film fizikai elemzésével megmondani, hogy merre vannak hibák, sérülések, így a digitális feldolgozásnál már „csak" javítani kell.

A filmzaj, azaz a celluloid filmen látható szemcsék kiszűrése – vagy éppen létrehozása – szintén nagyon fontos

feladat, bár ez a terület inkább a képfeldolgozás témaköréhez tartozik, és nem sokban kapcsolódik a gépi látáshoz. A probléma nehézségét elsősorban az jelenti, hogy az eredeti apró részletek – magas frekvenciájú jelek – megőrzésével kell a zajt kiszűrni a képből. A legjobb minőségű zajszűrő programok – sőt gyakran hardverek – éppen ezért nagyban támaszkodnak a mozgásvektorokra: összehasonlítva egy adott képkockát, valamint a szomszédos képek alapján előállított változatot, a különbség – elvileg – csak a zajban jelentkezik. Fontos megjegyezni, hogy a filmzaj létrehozása legalább annyira fontos, mint a szűrése. Egyrészről a filmkészítők gyakran mint művészi kifejezőeszközt, speciális effektust alkalmazzák, másrészről ez a videós anyagok „filmesítésénél" megkerülhetetlen.

Super-resolution

Bár a szórakoztató iparban – egyelőre – nem alkalmazzák, születtek ígéretes algoritmusok a mozgó- és állóképek felbontásának automatikus javítására. Két tipikus példa a biztonsági videorendszerek által rögzített – általában pocsék – képeken elrejtett részletek kinyerése, valamint az űrkutatás területén, a képrögzítő eszközök natív felbontását messze meghaladó, szupermagas felbontású képek létrehozása. Az utóbbi alkalmazási terület esetében általában mozdulatlan objektumokról készülnek a felvételek, így lehetőség van több képet készíteni, és ezekből generálni egy, nagyobb felbontású képet. Az eljárás alapja lehet több, egymást részben lefedő kép összeállítása egy panorámává, illetve egymást szinte tökéletesen fedő, csupán töredékpixelekkel elmozdított kép „összedolgozása". Mindkettő esetben a forrás-képek egymáshoz viszonyított helyzetének pontos ismerete a kulcs, hiszen ennek felhasználásával viszonylag egyszerű kiszámítani a végeredményt.

Eredeti kép

Hagyományos módszerrel nagyított kép

Speciális super resolution eljárással, többletinformáció nélkül nagyított kép

De mit tudunk tenni, ha csak egy, fix – pl. biztonsági – kamera rögzíti a képeket? Nem sokat, ha mozdulatlan a megfigyelt tárgy vagy személy. De gondoljunk csak a kamera előtt elhaladó autó rendszámtáblájára, vagy a sétáló bankrablóra egy megfigyelt parkolóházban. A korábban ismertetett módszerekkel le tudjuk követni a célpont mozgását, mégpedig akár a képpontok által meghatározott részletességnél sokkal pontosabban is. Ezt a mozgást kompenzálva máris kaptunk egy képsorozatot ugyanarról az objektumról, ahol az egyes képek – akárcsak a korábbi példában – egymáshoz képest csak töredékpixelekkel mozdultak el. Így lehetőség nyílik a videoszekvenciából egy, nagy felbontású képet generálni.

A legnehezebb eset persze kétségkívül az, ha csupán egy darab kép áll rendelkezésre, és azt szeretnénk felnagyítani. Persze csodák nincsenek: a képről hiányzó információt nem tudjuk utólag kitalálni! Akkor mégis mit tudunk tenni? A választ a tanuló algoritmusok adják meg. Ha – elegendő mennyiségű és tartalmú – példán keresztül megtanulja az algoritmus, hogy adott képdarabok hogyan néznek ki felnagyítva, ezt a tudást alkalmazva ismeretlen képeket is fel tudunk nagyítani. Persze fennáll annak a veszélye, hogy a feldolgozott kép egy része hasonlít a tanulóhalmaz egy elemére, mégis téves eredményt ad a megtanult nagyítás alkalmazása.

Érdemes itt megjegyezünk, hogy ez az algoritmus nem csak a felbontás javítására alkalmazható, hanem fekete-fehér képek kiszínezésére is! Ebben az esetben szürkeárnyalatos képek és színes megfelelőjük képezik a tanítóhalmazt, ami alapján az algoritmus megtanulja, hogy a különböző alakzatok, struktúrák milyen színűek. Sajnos ez a technika csak akkor fog meggyőző és hihető eredményeket adni, ha a tanuláshoz

Rossz minőségű videofelvételek alapján kiszámított nagy felbontású állóképek

használt képek hasonlóak a feldolgozandóra: erdőről készült képeket csak „erdős” képek alapján, felhős képeket pedig csak „felhős” képek alapján tudunk kiszínezni, illetve felnagyítani.

Mindezen megkötések – mind a super-resolution, mind az utólagos színezés esetében – adják a magyarázatot arra, hogy video- vagy filmfelvételek esetében miért nem alkalmazzák ezeket az algoritmusokat. Produkciós környezetben nem elfogadható az, hogy egy eszköz működik dinamikus jeleneteknél, de nem működik akkor, ha szinte semmi nem mozog a képen. Szintén igencsak problémás, ha egy régi film minden egyes jelenetéhez kell találnunk – nagyon – hasonlító színes képeket.

A cikkünkben bemutatott algoritmusok és eljárások jól illusztrálják, hogy jelenleg csak nagyon speciális „tudást” tudunk kinyerni a képekből, és viszonylag messze vagyunk még a képi tartalmak általános megértésétől. Ha végiggondoljuk, hogy milyen sok információ van egy képen vagy egy videofelvételen, beláthatjuk, hogy még van hová fejlődni. A gépi látás igen fiatal és dinamikusan fejlődő tudományterület, folyamatosan jelennek meg újabb és újabb eredmények. A mobiltelefonok mini-kamerákká válása, valamint az internetes video-megosztó oldalak népszerűsége valószínűleg katalizálja a digitális kép- és videofeldolgozás fejlődésének ütemét. Érdemes hát nyitottnak lennünk az új technológiákra, mert remélhetőleg hamarosan még jobban megkönnyítik a számítógépek a profi és amatőr videósok életét!

Vass Gergő

Tanítóhalmaz alapján (bal oldal) nagyított kép (jobb oldal) sokkal jobb minőségű, mint a hagyományos nagyítással nagyított

Tanuló algoritmusokkal a fekete-fehér képek és videók is kiszinezhetőek