|  |  |
| --- | --- |
| **Óbudai Egyetem**  **Neumann János Informatikai Kar**  **Alkalmazott Informatika Intézet** | oe_cimer_szines_print_res |

TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

**Meteorológiai észlelők támogatása gépi látó rendszerrel**

|  |  |  |
| --- | --- | --- |
|  | **Szerzők:** | **Bartha Márk** |
|  |  | mérnök informatikus BSc. szak, IV. évf. |
|  |  | **Simándi Gergely** |
|  |  | mérnök informatikus BSc. szak, IV. évf. |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | **Konzulensek:** | **Dr. Sergyán Szabolcs** |
|  |  | egyetemi docens |
|  |  | **Dr. Vámossy Zoltán** |
|  |  | egyetemi docens |

# Tartalomjegyzék

[Tartalomjegyzék 2](#_Toc385409413)

[1. Rendszer célja 4](#_Toc385409414)

[1.1. Észlelők munkájának támogatása 4](#_Toc385409415)

[1.2. Működtetés kényszerfeltételek mellett 5](#_Toc385409416)

[2. Borultság vizsgálata és felhőtípusok osztályozása 6](#_Toc385409417)

[2.1. Cél meghatározása 6](#_Toc385409418)

[2.2. Felhők típusai és jellemzői 6](#_Toc385409419)

[2.3. Borultság vizsgálata hibrid küszöböléses algoritmussal 9](#_Toc385409420)

[2.4. Borultság vizsgálata szaturáció méréssel 10](#_Toc385409421)

[2.5. Felhők osztályozása képfeldolgozással 13](#_Toc385409422)

[2.6. Eredmények kiértékelése 14](#_Toc385409423)

[3. Magassági szélirány meghatározása 17](#_Toc385409424)

[3.1. Cél meghatározása 17](#_Toc385409425)

[3.2. Előfeldolgozás 18](#_Toc385409426)

[3.2.1. Probléma meghatározása 18](#_Toc385409427)

[3.2.2. Gauss simítás 18](#_Toc385409428)

[3.2.3. A Piramis módszer – felbontás hierarchiák 19](#_Toc385409429)

[3.3. Jellemző pontok detektálása 20](#_Toc385409430)

[3.3.1. Jellemző pontok 20](#_Toc385409431)

[3.3.2. Sarokpontok megkeresése 20](#_Toc385409432)

[3.4. A felhők elmozdulásának meghatározása 21](#_Toc385409433)

[3.4.1. Az optikai áramlás 21](#_Toc385409434)

[3.4.2. Az elmozdulás azonosítása 21](#_Toc385409435)

[3.5. Eredmények kiértékelése 25](#_Toc385409436)

[4. Esőzés kezdetének és befejeződésének megállapítása 26](#_Toc385409437)

[4.1. Cél meghatározása 26](#_Toc385409438)

[4.2. Esőcseppek fizikai jellemzői 26](#_Toc385409439)

[4.3. Lehetőségek eső detektálására 27](#_Toc385409440)

[4.3.1. Esővonalak detektálása hisztogram vizsgálattal mozgókép sorozaton 27](#_Toc385409441)

[4.3.2. Esőcseppek felismerése gépjármű szélvédőjén 28](#_Toc385409442)

[4.4. Esőcseppek detektálásának folyamata üvegbúrán 29](#_Toc385409443)

[4.4.1. Előfeldolgozás a jól elkülöníthető cseppek detektálásához 29](#_Toc385409444)

[4.4.2. Előfeldolgozás morfológiai műveletek alkalmazásával 30](#_Toc385409445)

[4.4.3. Esőcseppek felismerése hibrid szegmentáló algoritmussal 30](#_Toc385409446)

[4.5. Kezdeti és befejeződési időpont becslése 31](#_Toc385409447)

[4.6. Elért eredmények 32](#_Toc385409448)

[5. Összegzés 33](#_Toc385409449)

[5.1. Elért eredmények 33](#_Toc385409450)

[5.2. Továbbfejlesztési lehetőségek 33](#_Toc385409451)

[Irodalomjegyzék 34](#_Toc385409452)

# 1. Rendszer célja

## 1.1. Észlelők munkájának támogatása

Az Országos Meteorológiai Szolgálat (továbbiakban OMSZ) észlelő állomásain – például Siófokon – manapság az időjárási jelenségek felmérése még becsléssel, az észlelők által történik. Ezeket a megfigyeléseket kihelyezett kamerák - például a meteorológiai "webkamerák" – segítségével is el lehetne végezni. Továbbá olyan helyeken is sikerrel alkalmazható a képfeldolgozás, ahol nincsenek állomások csak kamerák, így ott is becsléseket tehetnénk adatokra.

Projektünk az OMSZ nagyfokú támogatása mellett indult fejlődésnek. Hasznos szakmai információkat és tömérdek nagy felbontású képet szolgáltattak nekünk, amelyek nagyban segítették a munkánkat. Emellett sikerült szert tennünk egy, a kihelyezett kamerákkal megegyező felépítésű gépre, búrával és fűtő rendszerrel együtt, amivel a képek mennyiségét tovább növelhettük. Így esőben készült képekhez is sikerült hozzájutni. Mindezek mellé az észlelések során készített hivatalos dokumentum is elérhetővé vált számunkra, ami lehetővé tette a pontos tesztelést. Megerősítették, hogy ehhez hasonló rendszerrel még nem volt lehetőségük dolgozni, így rendkívül érdekesnek és hasznosnak tartják a munkánkat. Fontosnak találjuk megjegyezni, hogy az észlelők munkáját nem kiváltani szeretnénk, hanem segítségünkkel pontosítani a becsléseiket.

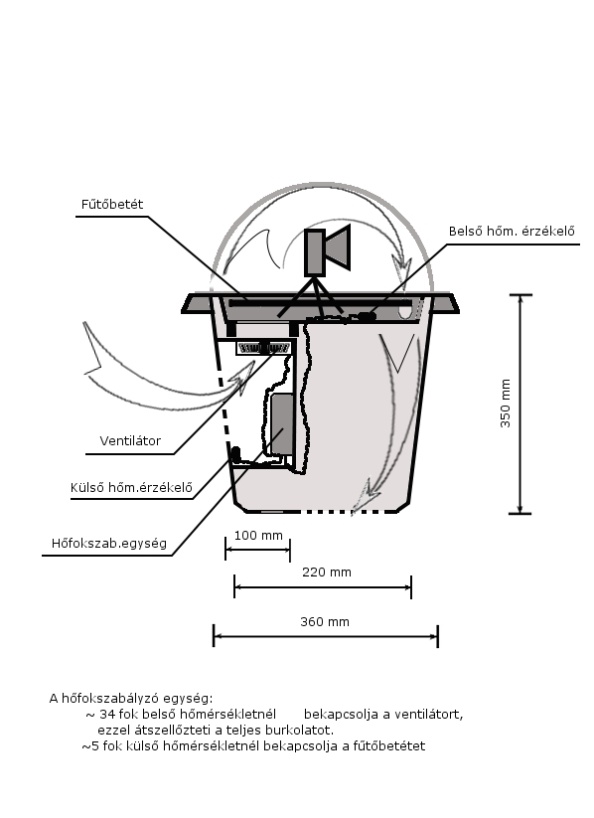
Meg kell említeni, hogy az OMSZ észlelései 1 órás időközökkel a nap 24 órájában történnek. A felhőzet vizsgálatát az észlelők teljesen a saját becsléseikre hagyatkozva végzik, ami általában pontatlan. Az emberi értékelés akkor is szubjektív, ha több fő vizsgálja ugyanazt a felhőzetet. Egy ismert probléma az ún. „packing effect”, ami azt jelenti, hogy az emberek túlbecslik a horizont közelében található felhők mennyiségét. A borultság megállapításában jelenleg a gépi rendszerek pontosabbak, míg a felhők típusának vizsgálatában az észlelők. Éppen ezért sok hasonló rendszer nem, vagy csak a statisztikák készítéséhez veszi figyelembe az emberi méréseket. Csapadék esetén pedig az észlelőket műszerek támogatják az állomásokon, így e meteorológiai tulajdonságokat pontosan tudják feljegyezni. Megállapítható, hogy igazán pontos eredmény az informatikai rendszerek és az emberi becslések együttes alkalmazásával kaphatunk.

A programot a kapott képek segítségével pontosítottuk, elemeztük a működést, és új funkciókkal egészítettük ki. Ezen kívül létrehoztunk egy közös felhasználói felületet az egyes részek kezelésére, így az egész rendszer könnyen átláthatóvá és kezelhetővé vált, és ilyen módon a külön "feladatok" fel is használhatják egymás eredményeit, amennyiben szükséges. Programunkkal tehát egy olyan rendszer elkészítését tűztük ki célul, amellyel meg tudjuk becsülni az ég borultságát, a magassági szél irányát, az esőzés kezdetét és végét, valamint a felhők típusait.

## 1.2. Működtetés kényszerfeltételek mellett

Az OMSZ [1] specifikációjában ismertette az észlelők munkája során alkalmazott eszközöket és módszereket, melyekhez nekünk alkalmazkodnunk kellett a rendszerünk fejlesztésekor. Jelenleg Nikon D50 – D60 – D70 kamerák vannak használatban, nagy látószögű objektívvel. A kamerákat burok veszi körül, ami védi a viharoktól, és egyedi fűtésrendszere van (lásd 1. ábra). A kamerák jelenleg 10 perces időközökkel készítenek 1-1 képet, viszont a programunk egyes funkcióihoz egymás után gyorsan elkészített képek szükségesek. Sajnos az OMSZ nem biztosítja számunkra azt, hogy a fényképezőgép mindig másodpercenként fényképezzen, mert ezzel a kamerák optikájának a kopási folyamatát gyorsítanánk meg. Erre azt a megoldást találtuk, hogy marad a 10 perces időköz, de akkor a kamera 3-5 képet készít pár másodperc eltéréssel. Így a rendszerünk összes funkciójához meglesznek a megfelelő képek.

A rendszer teljes működéséhez akár egy kamera is elegendő lenne, ám bizonyos funkciók akár két kamera esetén is megállják a helyüket. Ennek oka a rendszerünk sokrétűsége. Borultság, felhő típusok, és felhők mozgásirányának vizsgálata esetén egy nem függőlegesen beállított kamera képe hamis információt adna. Az irányokat tévesen becsülnénk meg, borultság vizsgálatánál pedig nem kapnánk pontos értéket, mivel nem tudnánk eldönteni, hogy hol látjuk a felhő függőleges illetve vízszintes részeit, továbbá az ég alulról nézve fedetlen részei is rejtve maradhatnának a kamera elől. Ebben az esetben a felhők detektálása is problémát jelentene, mivel oldalról nézve más a színezetük, így a detektálás bizonyos szakaszai nem lennének elvégezhetőek. Azonban esőzés kezdetének és végének detektálásakor olyan képek is szóba jöhetnek, amin még látszik a horizont, mivel ilyenkor az esőcseppek tükrözik a földet és az eget, és ez a határ segíti a felismerésüket.



1. ábra – Az OMSZ által használt kamerák szerkezeti felépítése.[1]

# 2. Borultság vizsgálata és felhőtípusok osztályozása

## 2.1. Cél meghatározása

A meteorológiai észlelők az ég borultságát is vizsgálják. Ennek osztályozása egy 0-8 oktás rendszerben történik. Ha az érték 8, akkor az eget teljesen elfedik a felhők, míg 0-nál teljesen felhőtlen. Amennyiben azonban az értéket 8 oktának számoltuk, de van kis lyuk a felhőzeten, az értéket 7 oktának kell venni. Ugyan így 0 oktánál, ha egy kis felhő van az égen, már 1 okta az értéke. Ezen kívül fontos, hogy amennyiben 8 oktás felhőzetünk van, de az Cumulus típusú, feltételeznünk kell, hogy a felhők között lyukak vannak még akkor is, ha ez nem látszik, és 7 oktának kell vennünk a borultságot. [1]

Ha a felhőket el tudjuk különíteni az égtől, akkor ez az érték már könnyen kiszámítható a kamerán látható területre vonatkozóan. A borultság vizsgálatot a meteorológiánál jelenleg az észlelők végzik ránézéses becslés alapján. Tehát az első feladat számunkra a felhők detektálása, égtől való elkülönítése, majd az imént említett feltételek mellett meg kell tudnunk határozni az ég oktában mért borultságát.

A felhők típusának megállapítása már egy összetettebb feladat. Míg a borultság vizsgálatban jelenleg a gépek pontosabbak, ebben a témakörben még az észlelők becsülik meg legpontosabb típust. Részletes megvalósítása rendkívül nehéz témakör, mivel sok felhőtípus létezik, melyek gyakran csak apró tulajdonságokban térnek el egymástól. Ezen tulajdonságok némelyike kép alapján nem is megállapítható (földtől mért távolság).

## 2.2. Felhők típusai és jellemzői

A felhőket magasság szerint 3 rétegbe, és ezeken belül 10 szintre oszthatjuk (lásd 1. táblázat). A szintek jelentik a különböző típusokat, de ha igazán részletesek akarunk lenni, akkor még ezeket is tovább oszthatjuk altípusokra. Az alábbi adatokat a [2] könyvben leírtak alapján és az [1] specifikációból gyűjtöttük össze.

Magasan lévő felhők a Cirrus, Cirrocumulus és a Cirrostratus. Ezekre egységesen igaz, hogy 6000m fölött helyezkednek el, jégkristályokból állnak és nem adnak csapadékot.

A Cirrus egy világos színű, szálas szerkezetű magasan elhelyezkedő felhő. Általában egyszerre csak kevés - néhány okta - van belőle az égen. Érdekesség, hogy gyakran a repülőgépek kondenz csíkja is Cirrus-szá alakul, ha sokáig megmarad (és ez nem baj, nem kell leszürni a felhők közül).

A Cirrocumulust a köznyelv "bárányfelhő"-nek nevezi. Szintén magasan helyezkedik el, gomolyos, általában fehér színű. Azonban nagyobb önárnyéka lehet, tehát nem csak fehér, hanem egész sötét színt is felvehet, tartalmazhat ezek között színátmenetet.

A Cirrostratus egy magasan elhelyezkedő egybefüggő fátyolfelhő. Képes az egész eget beborítani. Különleges ismertetőjele, hogy nagyon hasonlít az Altostratusra, de átlátszik rajta a Nap, és körülötte Halo jelenség alakul ki, tehát egy gyűrű látszik.

|  |  |  |
| --- | --- | --- |
| **Latin név** | **Magyar név** | **Szint** |
| [Cirrus](http://hu.wikipedia.org/wiki/Cirrus) | pehelyfelhő | magas |
| [Cirrocumulus](http://hu.wikipedia.org/wiki/Cirrocumulus) | bárányfelhő | magas |
| [Cirrostratus](http://hu.wikipedia.org/wiki/Cirrostratus) | fátyolfelhő | magas |
| [Altocumulus](http://hu.wikipedia.org/wiki/Altocumulus) | párnafelhő | középmagas |
| [Altostratus](http://hu.wikipedia.org/wiki/Altostratus) | lepelfelhő | középmagas |
| [Stratocumulus](http://hu.wikipedia.org/w/index.php?title=Stratocumulus&action=edit&redlink=1) | gomolyos rétegfelhő | alacsony |
| [Stratus](http://hu.wikipedia.org/wiki/Stratus) | rétegfelhő | alacsony |
| [Cumulus](http://hu.wikipedia.org/wiki/Gomolyfelh%C5%91) | gomolyfelhő | alacsony |
| [Nimbostratus](http://hu.wikipedia.org/wiki/Nimbostratus) | esőrétegfelhő | több szintet átfog |
| [Cumulonimbus](http://hu.wikipedia.org/wiki/Cumulonimbus) | zivatarfelhő | több szintet átfog |

1. táblázat – Felhők típusai.[2]

A középmagasan elhelyezkedő felhők 2500-6000m között találhatók. Típusai az Altocumulus, Altostratus és Nimbostratus.

Az Altocumulus egy gomolyos, változatos színű középmagasan elhelyezkedő felhő. Nem csak a színe, hanem az alakja is változik, csapadék általában nem származik belőle, kivéve az AC7-et, amiből gyenge csapadék származhat. Változó mennyiségben van az égen, 1-7 oktáig.

Az Altostratus is középmagasan van, és a teljes eget beborítja. Általában egysíkú, egyszínű, és nem ad csapadékot. Két fajtája van. Az egyiken átlátszik a Nap és Halo-ja (gyűrűje) van, a másikon nem látszik át.

A Nimbostratus egy vastag, egész eget beborító, középmagasságban elhelyezkedő felhő. Több napig tartó eső vagy hó származhat belőle. Vastagsága miatt a Nap nem látszik át rajta. Nagyon hasonlít az Altostratusra, ezért akkor mondhatjuk Nimbostratusnak egy felhő típusát, ha csapadék esik belőle. Egyedi jellemzője, hogy esőn és havon kívül fagyott eső is származhat belőle.

A Cumulus egy alacsonyan elhelyezkedő felhő. Nem boríthatja teljesen az eget, 7 oktánál nem lehet több. Az alja egyenes, de egy habos, fölfelé növő tetőrész indul ki belőle. Ha kellően magas, akkor záporszerű eső vagy hó származhat belőle.

A Cumulonimbus a Cumulus egy továbbfejlődött változata. Akkor nevezzük így a felhőt, ha a Cumulus magassága átlép a közepmagas szintre. Vastagsága miatt sötét az alja, teteje változatos színű, habos. Legnagyobb alakja úgynevezett üllővel rendelkezik. Ekkor a magassága már meghaladhatja a 6000 métert, és zivatar származhat belőle dörgéssel, villámlással, jégesővel, intenzív esővel vagy hóval.



2. ábra – A két felhőtípus, melyeket gépi látó rendszerrel meg lehet különböztetni. Balról a Cumulus, jobbról a Stratus látható.

A Stratocumulus alacsonyan elhelyezkedő párna alakú felhő. Sötét foltok vannak rajta, de a széle világos. Vertikális magassága kicsi, gyenge esőt adhat. Maximum 7 okta lehet belőle.

A Stratusnak két fajtája lehet: Virga és Stratus. A Virga csapadékos időben fordul elő, sötét, cafatos szélű, szálas szerkezetű, míg a Stratus a felszállt köd. Színe világos, egysíkú, egyszínű, és szitálást eredményezhet.

A felsorolt típusok gyakran csak apró, nehezen detektálható részletekben térnek el. Ilyen tulajdonság például, hogy látszik-e a Nap gyűrűje, a felhő magassága – amit az általunk használt képek alapján nem tudunk megállapítani – a felhő vastagsága és az, hogy a felhő milyen módon alakult ki, ami pedig bonyolult mozgókép feldolgozást igényelne. Viszont észrevehető, hogy bizonyos felhők főbb tulajdonságai megegyeznek. Ilyen módon a listát két csoportra bonthatjuk: Stratus és Cumulus típusú felhők (lásd 2. ábra). Az elsőbe tartozik többek között a Stratus, Cirrus, Nimbostratus, Altostratus, a másodikba pedig a Cumulus, Cumulonimbus, Altocumulus. Programunk első verziójában ezek felismerése lesz a cél.

A Stratus és a Cumulus között az egyik legnagyobb különbséget az égen található mennyiségük, tehát az ég borultsága adja. Így egyértelmű, hogy ez a témakör szorosan összekapcsolódik a Borultság vizsgálat témájával. A Cumulus jellegű felhők 1-7 oktát boríthatnak be, míg a Stratus jellegűek leggyakrabban 7-8 oktát. Azonban ez gyakran még nem elegendő információ ahhoz, hogy biztosan megállapíthassuk a felhő típusát, és a kameránk nem az egész eget figyeli, csak annak egy részét, aminek következtében Cumulus típusúra is kaphatunk 8 oktás értéket. Ezért figyelembe kell vennünk a felhők színezetét is. A Stratus mindig közel egyszínű, míg a Cumulus rendkívül változatos sötétségű lehet.

Ebből a két szempontból már megállapítható, hogy a képen látható felhők melyik csoportba sorolhatók. Azonban van még egy probléma. Előfordulhat, hogy nem tiszta az idő, de a Cumulus-os felhőzet közel 8 okta, és így a lyukak a felhők között nem látszanak. Ez a fenti két szempont együttes figyelembe vételével kiküszöbölhető, ugyanis hiába nem látunk lyukat, a szakirodalmak szerint ([1] és [2]) feltételezhetjük, hogy ott vannak, amennyiben a felhő színei több árnyalatot is felvesznek. Tehát a felhőnk Cumulus lesz, a borultsága pedig 7 okta.

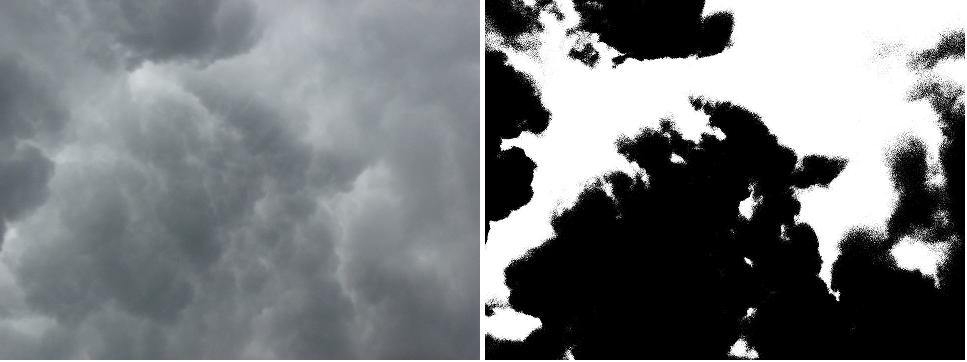
## 2.3. Borultság vizsgálata hibrid küszöböléses algoritmussal

Az elsőnek vizsgált lehetséges megoldás felhők detektálására a [3] műben ismertetett módszer. Ez egy olyan hibrid küszöbölést ismertet, amely az információ kinyerésre éldetektálást használ, küszöbölésre pedig a P-tile módszert. Ebben az esetben mindig szürkeárnyalatos képpel dolgozunk. Feltesszük, hogy az objektum világosabb a háttérnél. Ezen kívül azt is állíthatjuk, hogy az objektumok a kép egy bizonyos %-át elfoglalják. Ezt jelölhetjük P%-kal. A küszöbölő algoritmusunk úgy működik, hogy mindig legalább a P%-ot elérjük a végeredményben.

A teljes módszer másik fele az éldetektálás, ami segíti a képből való információ kinyerését. Az élek keretet adnak az objektum(ok) és a háttér között. Élkeresés eredményeként egy "edge map" jön létre, ami az éleinket tartalmazza. Az élkeresésnek számos algoritmusa ismert, de két típusba sorolhatók: gradiens és Laplace. Gradiensnél lokális maximum és minimum értékeket keresünk az első deriváltban, míg Laplace-nál zérus helyeket a második deriváltban. Nekünk egy jó élkeresőre van szükségünk, tehát a következő feltételeknek kell megfelelnie: lehető legtöbb helyes élet adja, az élek a lehető legközelebb legyenek a pontos helyükhöz, minden élet csak egyszer találjunk meg, a zaj pedig ne keltsen hamis éleket.

Egy számunkra megfelelő algoritmus a Canny éldetektálás, mivel ez zajszűrést is végez éldetektálás előtt. Persze más algoritmusok is használhatók, de minden esetben célszerű egy zajszűréssel kezdeni. Az egyik legismertebb és leggyakrabban használt ilyen szűrő a Gauss szűrő.

Ha sikerült a megfelelő éldetektálást végezni, akkor megkaphatjuk a P% értékét. Ennek módja az, hogy az eredeti kép "edge map"-jéből kivonjuk a küszöbölt kép "edge map"-jét. Ha ezt minden küszöbölő értékre elvégezzünk, akkor a kapott értékekből megkaphatjuk a P% értéket ott, ahol ez a különbség a legkisebb volt. A két kép kivonását az MSE- vel (Mean Squared Error) végezzük el.



3. ábra – A hibrid küszöbölést és éldetektálást alkalmazó algoritmus hibás működése változatos színű felhők esetén. Látható, hogy teljesen borult ég esetén is talál jóval a 0 és 255 közötti intenzitásértékek között is küszöbértéket.

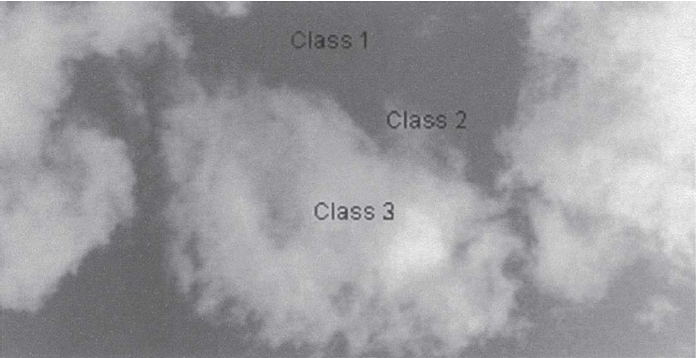
A Step érték változtatásával növelhetjük, illetve csökkenthetjük a pontosságot. Ha nagy értéket adunk neki a programunk gyorsabb, de pontatlanabb lesz, míg kis érték esetén pontosabb, de lassabb. A mi esetünkben a pontosság előbbre való a sebességnél, mivel a kép feldolgozására kb. 10 perc áll a rendelkezésünkre az újabb kép készítése előtt. A hibrid algoritmus a tesztesetek többségében pontosabb eredményt adott az Otsu binarizálásnál.

Ezzel a megoldással azonban nem sikerült pontos eredményt elérni. Az első probléma, hogy mindig keres küszöbértéket, míg a mi esetünkben felhőtlen, vagy teljesen felhős ég esetén az is előállhat, hogy csak előtér vagy háttér van, tehát nincs szükségünk küszöbölésre. Ezen kívül egyetlen felhő is sokféle színárnyalatot felvehet egyszerre, így ilyen esetekben ez az algoritmus a felhőt is több objektumra bontja (lásd 3. ábra). Ezen kívül gyakori probléma, hogy küszöbölés után a változó színek miatt nem mindig a felhő lesz az előtér az ég pedig a háttér, ami lehetetlenné teszi a borultság számítását. Így rájöttünk, hogy hagyományos küszöbölő módszerek segítségével nem tudunk eredményt elérni. Új megoldásra volt szükségünk.

## 2.4. Borultság vizsgálata szaturáció méréssel

Létezik egy módszer, amely kifejezetten felhők detektálására lett kifejlesztve, ez pedig a szaturáció méréséből való következtetés. Ez a módszer a [4] cikkben került ismertetésre, amely alkalmazásával igen nagy előrelépést értünk el. E szerint a felhők dinamikusan változó rendszerek magas fényvisszaverő képességgel, és kék-piros színtartományba eső színekkel, általában fehérek. Ezzel szemben az ég színe a háttérben a zöld-piros színskálán mozog, általában kék, és statikusnak tekinthető. Ez az algoritmus a kamera által rögzített kép különböző hullámhosszait figyeli.

Az RGB modell széles körben elterjedt. Ez a színek és a szaturáció egy konstans fényerőn vett reprezentációja. Az IHS rendszerben az intenzitás (I) a teljes energiát jelöli az összes hullámhosszon, ami eléri a szemet. Ez felelős a fényerő érzékeléséért. A "Hue" (H) a fény elnyelését, visszaverődését adja meg, így ez felelős a színekért. A szaturáció (S) a színek tisztaságát jelöli. A magas szaturációs értékekre azt mondják, hogy tiszta, mint a derült ég. Az alacsony értékűek olyanok, mint a felhők.



4. ábra – Osztályok a pixelek besorolására. Class 1: tiszta ég; Class 2: nem meghatározott; Class 3: felhő [4]

|  |  |  |
| --- | --- | --- |
| **Változó** | **Paraméter** | **Érték** |
| Fa | Felhő átlag értéke | 12.7 |
| Ff | Felhő átlag tartománya | 3.7 |
| Ea | Tiszta ég átlag értéke | 45.3 |
| Ef | Tiszta ég átlag tartománya | 4.4 |

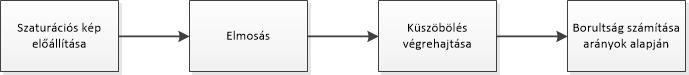
2. táblázat – Szakirodalomban meghatározott küszöbértékek a felhődetektálás során alkalmazott osztályoknál. [4]

Láthatjuk, hogy a felhőknek jó a fényvisszaverő képessége, általában fehérek, de számos színárnyalatot felvehetnek. Ezzel szemben a kék színű derült ég magas szaturációt eredményez. Ezért az IHS rendszerből mi a szaturációt fogjuk használni algoritmusunk alapjául. Kiszámításának egy lehetséges módja:

*S = 1 - (3 / (R + G + B)) \* min(R,G,B)* (1)

3 osztályt kell meghatároznunk a pixelek besorolására szaturációs értékük alapján (lásd 4. ábra). Minden osztályt egy alsó és felső küszöb értékkel határolunk. Ezek után a pixeleket a megfelelő értékek alapján a megfelelő osztályokba sorolhatjuk. Ezeket az értékeket más, a témával foglalkozó emberek megállapították, több mint 40 különböző kép feldolgozásával (lásd 2. táblázat). Az algoritmust meteorológusok segítségével tesztelték. Képeket adtak nekik, amiken meghatározták a borultságot, majd a kapott értékeket összehasonlították a program által számolt értékekkel. Végeredményként az algoritmus 94%-os pontosságot eredményezett tiszta ég, és 99%-osat felhős ég esetén.

Az algoritmusban (lásd 5. ábra) szaturáció számítására több képletet is kipróbáltunk, de a fent említett adta a legtisztább eredményt. Más módszereknél gyakran erősen ki lettek emelve az élek, amire nekünk jelenleg nincs szükségünk. Azonban a fent definiált értékek alkalmazásával a különböző szintek határaira nem kaptunk pontos eredményt. Ezen kívül figyelembe kell venni ezen a téren azt is, hogy nem fogja minden felhasználó azonos körülmények között, azonos típusú fényképező géppel alkalmazni a programot. Így célszerű egy konfigurálási megoldást bevezetni, hogy ezeket a határokat gyorsan és pontosan beállíthassuk használat előtt. Erre egy manuális és egy automatikus megoldást készítettünk. Automatikusnál a program előállítja a szaturációs képet (célszerű olyan képen végezni a konfigurálást, amin a felhő és ég jól elkülöníthető), ezen egy Otsu binarizálást végez, ami megad egy küszöbértéket, majd ez alapján az érték alapján 2 határt ad meg. Az ég felső- és a felhők alsó korlátját kihagytuk, mivel még elmosás után is, a zajok miatt a képen gyakran hibás értékeket kaptunk. Ez a megoldás gyors, és pontos határértékeket ad. Azonban ha azt mégis pontatlannak találjuk, lehetőségünk van manuális beállításra. Ebben az esetben egy új ablakban maximum 3 képet nyithatunk meg, melyeken 2 slider és valós időben végzett küszöbölés segítségével állíthatjuk be a határokat. Így mindig jól látható a változás, és egész pontosan be tudjuk állítani a küszöbértékeket a nekünk megfelelőre.



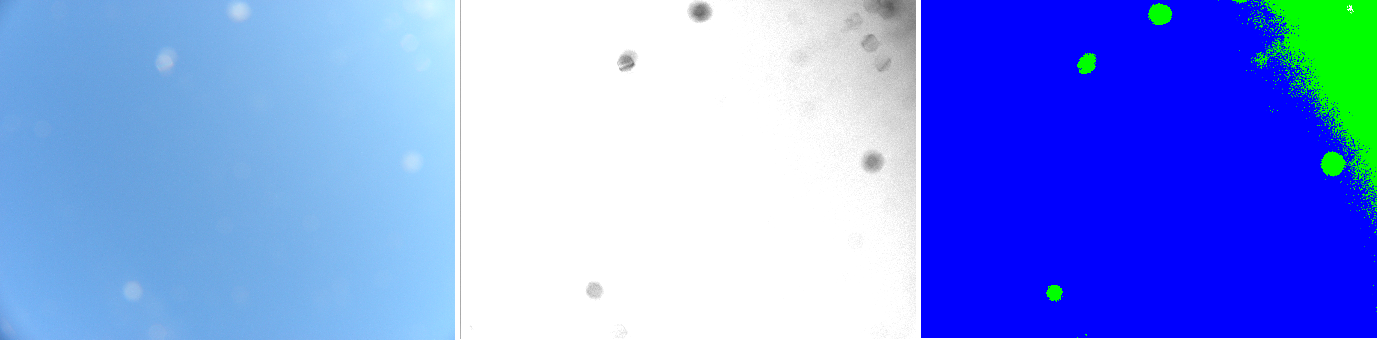
5. ábra – Borultság vizsgálat folyamata

Megfelelő konfigurálás után már jó pontosságot érhetünk el az algoritmussal, és könnyen kiszámíthatjuk a pontos borultságot. Ennek módja a küszöbölés utáni 3 szint arányainak összehasonlítása, majd az érték számítása százalékban, és átváltása nyolcadokra. Általános esetben a "Nem meghatározott" tartományba eső részeknek 0.5-ös súlyt adtunk. Azonban felhőtlen ég esetén előfordulhat, hogy a Nap által megvilágított fénylő részeket a program ebbe a tartományba sorolja, és így 0 helyett 1 oktát kapunk eredményül. (lásd 7. ábra) Ennek kiküszöbölésére a program először a "Nem meghatározott" típusú részek mellőzésével számolja ki a borultságot. Ha ez az érték 0, és az ég nem tartalmaz egy bizonyos kis értéknél nagyobb mennyiségű felhőnek detektált részt, akkor a borultság 0. Ellenkező esetben az algoritmust újra lefuttatjuk, de ezúttal a "Nem meghatározott" részeket is figyelembe véve. Ezzel a módszerrel az egész nagy részben megvilágított képeken is sikerült elérni, hogy helyes értéket kapjunk.

A fent leírt módszerrel biztosítottuk, hogy ha 0 oktás a felhőzet, de van rajta egy kevés felhő, akkor 1 oktának vegyük. Azonban hasonló módon 8 oktás felhőzetnél 7 oktának kell vennünk, amennyiben egy kicsit látszik az ég. Így az első futtatásnál ezt is le kell tesztelnünk. Ha 8 oktát kaptunk, és nincs ég a képen, akkor ezt az értéket elfogadhatjuk, de ha ez nem teljesül, akkor újra futtatjuk az algoritmust. De míg a 0 oktás feltételt második futtatásnál már nem kell vizsgálnunk, mivel pont a "Nem meghatározott" részek szűrése volt a cél, ebben az esetben ezt újra figyelembe kell vennünk, mivel a korábban nem vizsgált részekről, még nem tudjuk, hogy ég-e.



6. ábra – Eredeti kép, szaturációs kép, küszöbölt kép



7. ábra – Felhőtlen ég esetén zajt tapasztalhatunk, melynek kezelésére külön figyelmet kell fordítanunk.

## 2.5. Felhők osztályozása képfeldolgozással

A felhők sikeres detektálásából és a mért borultságból kiindulva próbáltuk meg megállapítani a felhők típusát is (lásd 8. ábra). Első lépésként az eredeti színes képet szürkeárnyalatossá kell tennünk, majd ezt maszkolnunk a küszöbölt képpel. Ezek után az eredeti képet vizsgálva egy hisztogramot kell készítenünk, de csak azokat a részeket vizsgálva, ahol felhő található, majd össze kell számolnunk, hogy hány intenzitás értéket találtunk. Ezeken az értékeken célszerű egy szűrést alkalmazni, hogy a zajokat kiszűrjük. Ezt úgy tettük meg, hogy csak bizonyos mennyiségű pixelt tartalmazó értékeket vettük figyelembe. Innentől kezdve minden szükséges információ a rendelkezésünkre áll a típus meghatározásához. Több kép vizsgálatával megállapítottunk egy küszöbértéket, ami felett a felhő Cumulus, alatta pedig Stratus. Ez lesz az elsődleges szempontunk a vizsgálatnál. Ha ez alapján megállapítottuk a típust, meg kell vizsgálnunk azt is, hogy a kapott érték megfelel-e a korábban kiszámított borultságnak. Amennyiben nem, annak megfelelően módosítanunk kell. Ezt úgy tesszük, hogy ha színárnyalatok tekintetében a felhő teljesen egyértelműen egy adott típusba sorolható, akkor nem módosítunk az értéken, míg ha nem teljesen egyértelmű a kapott érték, akkor az oktának megfelelően módosítunk.

A hibás kapott értékek közül számos abból adódik, hogy a kamera képe nem fedi le a teljes égboltot, míg az észlelők annak egészét vizsgálják. Ahogy már említettük, borultság meghatározásában jelenleg a gépek pontosabbak, így a kép alapján megállapított értékek többségében megegyeznek a program által számított értékekkel, de a teljes ég vizsgálatával készített hivatalos dokumentumok adatai a képek alapján megállapítottaktól gyakran eltérnek. Ennek javítására készíthetnénk egy rendszert, ami az egész eget figyeli, de célkitűzéseink között a jelenleg is használt kamerákkal való megvalósítás szerepelt. Tehát algoritmikus megoldás kell. Mivel más funkciókhoz sorozatképek szükségesek, így célszerű ebben az esetben is több képet vizsgálni egy helyett. A kamera mozdulatlan, de a felhők folyamatos mozgásban vannak a fényképezés ideje alatt. Ilyen módon hasonló hatást érünk el, mintha a kamerát mozgatnánk, hogy az égbolt nagyobb részét rögzíthessük. Természetesen így az eredmény függ a felhők mozgásának sebességétől, a képek készítésének gyakoriságától, és az első és utolsó felvétel között eltelt időtől. Az algoritmus végig fut az adott sorozaton, minden képre egyenként megállapítja a borultság értékeket, majd ezeknek az átlagát veszi. A típusok közül a legnagyobb mértékben előfordulót választja. Habár a teljes égbolt még így sincs vizsgálva, nagyobb területet figyelhetünk, mint egyetlen képpel, és csökkenthetjük a véletlenszerű zajok miatti hibás detektálás valószínűségét.

A kamerák a nap 24 órájában készítenek képeket, de az éjszakai képeken felhők már nem láthatók a sötét miatt. Így a programunkban ezt az esetet is le kell kezelnünk. Erre megoldásnak egy algoritmust készítettünk, mely a kép intenzitás értékeinek átlagát vizsgálja. Számos képet megvizsgálva azt figyeltük meg, hogy 8 oktás, teljesen sötét felhőzet esetén is jelentősen magasabb az intenzitások átlaga, mint esténként. Így megállapítottunk egy értéket, amin még sikerült a felhőket detektálni, de ami alatt már nem lehetséges. Ezt a vizsgálatot a program minden detektálás előtt lefuttatja, és amennyiben úgy ítélte meg, hogy a detektálás nem lehetséges, leállítja a folyamatot és ezt jelzi nekünk.

Felmerült ezen kívül a kérdés, hogy mi történik akkor, ha esik az eső. A kamerát egy búra veszi körül, amin az esőcseppek jól látszanak, ez zavarhatja a méréseket. Ebben az esetben az eső detektálására készített algoritmus felhasználásával megítélhetnénk, hogy esik-e az eső, és amennyiben igen, a felhők detektálását nem futtatjuk le. De a tesztképek alapján azt az eredményt kaptuk, hogy habár az eredeti képen az esőcseppek nagymértékben látszanak, szaturációs kép és elmosás után szinte teljesen eltűnnek, és így pontos eredményt kaptunk.

## 2.6. Eredmények kiértékelése

Felhős ég esetén az algoritmus rendkívül pontosan képes az eget elkülöníteni a felhőktől, legyen szó akár sötétebb, akár világosabb félékről. Ezen kívül teljesen borult ég esetén is képes megmondani, hogy a képen csak felhő látható, és felhőtlen égbolt esetén is pontos értéket ad. Ennek megfelelően a borultságra is jó értéket kapunk. Fontos megjegyeznünk, hogy a meteorológiai észlelők 1 oktás hibahatárral dolgoznak, bármikor előfordulhat, hogy 1 oktát tévednek a becslésekben. Ezért a mi algoritmusunk tesztelésénél is az 1 oktás eltérést még jónak számítottuk!

A programot több típusú felhőn, teljesen borult és felhőtlen égen egyaránt teszteltük különböző körülmények között. Ezek alapján az látszik, hogy viszonylag nagy mértékig kiszűri a zavaró tényezőket, és jó aránnyal helyesen állapítja meg a típust. Azonban figyelembe kell vennünk, hogy a képen egyszerre több típusú felhő is előfordulhat. Ebben az esetben a program jelenleg még nem képes megállapítani a különböző típusokat. Mivel ha van jelentősebb Cumulus, akkor magas lesz az intenzitások száma, a program ilyenkor ebbe a típusba sorolja a felhőzetet annak ellenére, hogy a háttérben található-e Stratus.



8. ábra – Felhő típus detektálásának folyamata

Az OMSZ képei közül a hivatalos észlelések dokumentumai alapján több száz képen megállapítottuk a felhő típusát, és a borultságot. Ezt az adatot előre meghatározott módon (Típus\_Sorszám\_Borultság.jpg) a képek címébe foglaltuk. A képek között szerepel sorozat is, de véletlenszerűen készített képek is készültek. A programban készítettünk egy funkciót, mely több képen egymás után képes elvégezni a detektálást, és a kapott értéket összehasonlítani az észlelők által mértekkel, és a méréseket egy szöveges állományban rögzíti. Ezen kívül külön funkció készült a sorozatképekkel való tesztelésre, ahol beállíthatjuk, hogy hány egymás utáni képet tekintünk egy sorozatnak, majd a futtatás után az eredményt hasonló módon tárolja.

A program az elmúlt időszakban a nagyobb mennyiségű tesztképnek köszönhetően jelentős fejlődésen ment keresztül. Ez részben újfajta elmosás alkalmazásának köszönhető, részben pedig a régebbi algoritmus hibák kijavításának, amik bizonyos esetekben elrontották az értékeket. A program fejlődését egy diagramon mutatjuk be (lásd 9. ábra).

Az elért eredmény 491 képen tesztelve 87.06% lett, ami sorozatot és önálló képeket egyaránt tartalmaz. Ezt az értéket úgy kaptuk, hogy azokat a képeket melyeknél csak az egyik érték helyes, 0.5-ös súllyal számítottuk. A teszteket külön-külön is elvégeztük sorozaton és egyedi képeken, súlyozás nélkül (ha egy érték is hibás, akkor teljesen rossznak számítjuk a képet). Az eredményt az 9. ábra tartalmazza. Korábban említettük az esős képek kérdését. A tesztek alapján az ilyen képek 85.71%-a helyesen lett detektálva.

A hivatalos dokumentumokban a mérések 30 percenként vannak bejegyezve, de a képeink gyakran ezen időpontok között készültek, így a beállított értékek bizonyos esetekben egyértelműen eltérnek a készítés időpontjához legközelebbi bejegyzett adattól. Ez alapján megvizsgáltuk újra a képeket, és fals negatív, illetve fals pozitív csoportokba is besoroltuk őket. Az eredményeket az 3. és 4. táblázat tartalmazza.

. ábra – A programot véletlenszerűen készített, nagyban eltérő képeken, és sorozaton is teszteltük. A diagram a nem súlyozott értékekkel való számítás eredményét ábrázolja.

. ábra - A program a nagy mennyiségű tesztképnek köszönhetően sokat fejlődött az első működő verzió óta. Az ábra ezt a fejlődést szemlélteti az új sorozaton végzett tesztek alapján, a program első verziójából kiindulva a jelenlegiig. Az értékeket súlyozva állapítottuk meg.

|  |  |  |
| --- | --- | --- |
|  | Képen megadott borultság érték helyes | Képen megadott borultság érték helytelen |
| Borultság érték helyes | 471 | 0 |
| Borultság érték helytelen | 20 | 0 |

. táblázat - A mérések eredménye borultság tekintetében, fals pozitív és fals negatív (jobb oldali oszlop) értékeket is figyelembe véve, sorozaton és különálló képeken együtt tesztelve

|  |  |  |
| --- | --- | --- |
|  | Képen megadott típus érték helyes | Képen megadott típus érték helytelen |
| Mért típus helyes | 394 | 1 |
| Mért típus helytelen | 89 | 7 |

. táblázat - A mérések eredménye felhő típus tekintetében, fals pozitív és fals negatív (jobb oldali oszlop) értékeket is figyelembe véve, sorozaton és különálló képeken együtt tesztelve

# 3. Magassági szélirány meghatározása

## 3.1. Cél meghatározása

Az OMSZ szeretné, ha felhőtípusok meghatározása után a felhők mozgásának az irányát is megtudnánk határozni. A rendszerünk következő mérföldköve tehát így nem más, mint az objektumok követése és mozgásainak meghatározása. A kinyert adatokból szeretnénk megbecsülni, hogy az adott időpillanatban az idő szélcsendesnek mondható-e vagy sem. Ha nem, akkor meg kell tudnunk határozni a magassági szél irányát. A felhők mozgásának nyomon követése több mint egy egyszerű objektum követés, ugyanis az általunk vizsgált objektumok nem merev testek, mivel a 3D-s környezetről 2D-s leképzést kapunk, mely információ vesztéshez vezet. Elmondható, hogy a felhők területe, kerülete, formája és még a mozgása sem biztos számunkra, ezért a probléma igen komplexnek mondható, arról nem is szólva, hogy a felhő elmozdulása nem garantálható, hogy lineáris lesz. Szerencsénkre sikerült az OMSZ észlelőivel megegyezésre jutni abban, hogy számunkra 3 képből álló sorozatképeket is készítenek 10 percenként, így ilyen időközönként megfelelő pontosságú információt szolgáltathatunk. Először is feltételezzük, hogy a 3 db elkészült képen az ~ 5 mp-es késleltetések alatt a felhők mozgásának iránya és sebessége lineárisnak mondható és a felhő homogenitása nem, vagy csak minimális mértékben változik. Másodsorban a szél irányának meghatározása leginkább aszerint a felhő szerint történik, amelyik elmozdulása a legtöbb információt tartalmazza számunkra.

Az OMSZ eddig két sorozatfelvételt készített nekünk, így ezeken a felvételeken kell megfelelő eredménnyel detektálnunk az elmozdulásokat. Egyik sorozatban Cumulus típusú felhők láthatóak, melyek 800-1000 méter magasságban helyezkednek el, míg a másodikon statocumulusok, melyek kb. 1500-1800 m magasságban vannak. A magassági szél mind a 2 szinten kb. 15-16-17 m/s, ám a sebesség meghatározása nem elvárás.



11. ábra – Sarokpontok detektálása felhős képen előfeldolgozó algoritmus futtatása előtt (balra) és után (jobbra).

## 3.2. Előfeldolgozás

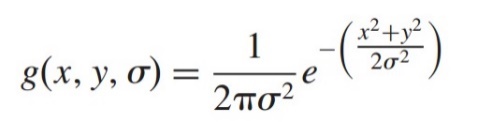
### 3.2.1. Probléma meghatározása

Ahhoz, hogy a felhők elmozdulásának az iránya megbecsülhető legyen, a kapott képrészleteket meg kell feleltetnünk egymásnak, majd ezáltal kell a mozgás irányát detektálnunk. Először meg szeretnénk határozni a képen a felhőket, így az elsődleges célunk, hogy az általunk kívánt objektumokon olyan jellemzőket keressünk, amelyeket utána a következő képen meg tudunk találni, majd megfeleltetni egymásnak. Ennek alapján pedig az elmozdulás iránya is meghatározható lesz, amiből már tudunk következtetni a magaslati szélnek az irányára is. A jellemző pontok megtalálására sarokpont detektort használunk, majd a pontok elmozdulását optikai áramlás segítségével fogjuk követni, így az összes pixel megfeleltetés helyett egy kisebb halmazt kell nyomon követnünk. A felhők mozgásának a detektálásához megbízható jellemzőket kell keresnünk, amiket majd követni tudunk a képsorozaton. Ezek a pontok tipikusan sarokpontok, amit a rákövetkező képeken próbálunk összepárosítani egymással.

Az eredeti képeink 11,8 MP-es felbontásban érkeznek, ahol  feldolgozási idő hosszadalmas lehet, és zajokat is tartalmazhat, így mielőtt bárminek is nekikezdenénk egy előfeldolgozási folyamatot kell végrehajtanunk a képeinken. Ha ezt nem tennénk, akkor nem megfelelő helyeken detektálnánk sarkokat. Mivel a legtöbb pont a kék homogén környezeten helyezkedik el, nem lehet elkezdeni a megfelelő vizsgálatokat. Előfeldolgozás után már sokkal jobb eredményt hozott számunkra az algoritmus. Ilyen esetekben a jellemző pontjaink inkább a felhők széleinél helyezkednek el mintsem homogén környezetben, ugyanazon beállítások mellett (lásd 11. ábra).

### 3.2.2. Gauss simítás

Az eredeti képünk magas zajterheltsége miatt kaptunk pontatlan sarokpont detektálásokat. Zajoknak nevezzük azokat, amelyek a képen egy oda nem illő, a környezetébe bele nem simuló intenzitású pixel(ek). Megszűntetéséhez átlagolhatunk, ha a kép minden egyes p pixelének egy kis környezetében kiszámoljuk az átlagot, majd ezt az értéket kapja meg a p. Ennél a módszereknél tudnunk kell, hogy a zaj mértéke ugyan csökken, de nem garantált, hogy sikeresen megszűntettük a zajt. Átlagoló szűrő helyett medián szűréssel is próbálkozhatunk, azaz ha a kis környezetben a statisztikából ismert medián értéket helyettesítjük a p pixel helyére. Ez a módszer sikeres lehet, ugyanis a medián nincs terhelve a zaj mértékével, viszont elmossa az éleinket. [5] A következő simító módszer a Gauss szűrő, ahol a számításnál a p pixeltől való távolságot is figyelembe vesszük. [6] Az előző szűrők gondja az volt, hogy ha növeltük a maszkunk méretét, akkor a p pixel nagyban különböző értéket vesz zajtalan képrészleteken is. A Gauss szűrő viszont súlyozott átlagolást fog adni, ahol a súly a távolsággal arányos. A szűrő számítási módszerét az (2)-es képletben láthatjuk.

 (2)



12. ábra – A képpiramis elve

Az általunk használt képeken egy elősimítást alkalmazunk, amit egy Gauss szűrő segítségével valósítottunk meg Azonban ez sem vezetett korrekt eredményhez. Egy alap 3x3-as maszkkal elsimított kép is elég sok zajjal van terhelve. A maszk növelése vezethet megoldáshoz, ám egy 13x13-as maszkkal végzett művelet is elfogadhatatlan detektálásokat tartalmaz. A 15x15-ös megvalósítás már a homogén felületen eltűntette a téves pontok észlelését, de a felhők széleinél nem talált pontokat, ami a vizsgálataink továbbhaladása szempontjából nem bizonyul elégségesnek. Ezért egy másfajta módszert kell alkalmaznunk.

### 3.2.3. A Piramis módszer – felbontás hierarchiák

Mivel a képeink mérete nagy és szeretnénk gyors algoritmusokat készíteni rá, így a megfelelő simítás mellett szeretnénk, ha tömörebb reprezentációba tudnánk leképezni őket. Erre az úgynevezett „Piramisok – felbontás hierarchiák” szolgálnak. [7] A kép különböző skálázású másolataiból épül fel egy kép, ami szintenként kettő hatványai szerint csökken vagy nő (lásd 12. ábra).

Mi a nagyobb képből egy kisebb felbontású képet szeretnénk készíteni, ezért a piramisban lefele lépkedünk. Ahogy az eredeti képünkön 3 szintet lépünk lefele, a képünk simább lesz és a felbontása is csökken, ugyanakkor a számunkra fontos pontokat tudjuk rajta detektálni és a homogén felületen se tapasztalunk véletlen sarokpont detektálásokat (lásd 13. ábra). Az algoritmusaink ideje is jelentősen nőtt.



13. ábra – Az előfeldolgozás folyamata: balra az eredeti kép, majd jobbra haladva eggyel csökkenő szintű piramisképek. Jobbszélső képen a 3. szintű piramiskép található.

## 3.3. Jellemző pontok detektálása

### 3.3.1. Jellemző pontok

A sarokpontok keresése nem más, mint élek találkozásának detektálása. Ha találtunk egy ilyen pontot, ott a képfüggvény több irányban változik. Ha csak egy élt találtunk, a változás csak egy irányba mutat. Homogén felületen nem beszélünk változásról. A sarokpontok figyelése emellett azért bizonyul jó megoldásnak, mert geometriai tanformációk mellett is megbízhatóan detektálhatóak, és nem kell minden képpontra számítást végeznünk majd. A célunk tehát olyan pontok megtalálása, amit az optikai áramlás során megbízhatóan fel tudunk használni. A sarokpontok detektálására több módszer is létezik. Az alapelvük az élek találkozásának megtalálása, azaz olyan ablakok keresése, ahol azok elmozdulása nagy differenciához vezet.

### 3.3.2. Sarokpontok megkeresése

Miután a képeinken a megfelelő előfeldolgozásokat végrehajtottuk, a jellemzők megkeresése következik. A megtalálásukhoz az ún. Good feature to track algoritmust fogjuk használni.

A követési eljárásban [8] tekintsük a megfelelően kiválasztott, követni szánt u pontot az I képen, és keressük hozzá a v pozíciójú pontot a másik J képen. Ebben a részben azt mondjuk meg, hogy mit jelent az első lépésben u meghatározása az I képen. Ezt a lépést jellemző-kiválasztásnak nevezik. A követés központi lépése az optikai folyam vektor kiszámítása. Itt a térbeli gradiens mátrixból képzett G mátrixról elvárjuk, hogy invertálható legyen, vagy más szóval G kisebbik sajátértékének is elég nagynak kell lennie (egy küszöbnél nagyobbnak). Ez jól jellemzi azt a pixelt, amit követni szeretnénk. Ezért a kiválasztás menete a következő [9]:

1. Számítsuk ki a G mátrixot és ennek kisebbik λm sajátértékét az I kép minden pixeléhez.
2. Legyen λmax a λm értékeknek maximuma a teljes képen.
3. Tekintsük azokat a pixeleket, amelyeknek a λm értéke nagyobb, mint a λmax érték bizonyos százaléka.
4. A megmaradt pixelekből tartsuk meg azokat, amelyeknek egy lokális környezetben a legnagyobb az értékük.
5. Végül csak azokat a pixeleket tartsuk meg, ahol a minimális távolság nagyobb, mint egy adott küszöbérték.

Az algoritmusnak átadhatunk egy Harris paramétert [10], így ezt az értéket fogja használni a sarokpontok detektálásához. Ha a paramétert elhagyjuk, akkor a minimális sajátérték keresése alapján fogja a pontjainkat megtalálni. Az algoritmus eredménye után a megmaradt pixelek lesznek a számunkra jól követhetőek (lásd 14. ábra). A következő részben ismertetett módszernek az inputjait fogják szolgálni a megtalált jellemzők.



14. ábra – Sarokpont detektorok alkalmazása felhős képen. Balra a Harris-féle sarokpont detektor, jobbra a minimális sajátértékkel számolt módszer.

## 3.4. A felhők elmozdulásának meghatározása

### 3.4.1. Az optikai áramlás

Az Optikai áramlás meghatározása nem más, mint több képen azonos képrészletek megfeleltetése és ezáltal az elmozdulás detektálása és/vagy sebesség meghatározása.  
A detektált elmozdulás nem feltétlenül egyezik meg a valós elmozdulással, mivel a valós háromdimenziós világot kétdimenziósra kell leképeznünk. Így ha nem tudjuk, hogy a kamera rögzített-e vagy sem nem lehet meghatározni, hogy a tárgy vagy a képfelvevő rendszerünk tett-e mozgást. Ugyanis ha  az objektum és a felvevő helyzete egymáshoz képest megváltozik, akkor az egy kétdimenziós képen a tárgy vetületének az elmozdulását jelöli. [11]

Feltételezzük, hogy a pixelek nem változtatják meg az intenzitásukat az elmozdulás alatt, hogy a kép térben koherens és időben állandó (egy felületi elem elmozdulása időben lassan változik). Egy probléma az úgynevezett apetúra probléma, azaz hogy az élre párhuzamos elmozdulást nem tudjuk detektálni, csak a merőlegeset.

### 3.4.2. Az elmozdulás azonosítása

Az általunk használt Piramistechnikával bővített Lucas–Kanade-féle jellemzőkövetés fogjuk használni az elmozdulás irányának meghatározásához.

A probléma megoldása érdekében a klasszikus Lucas–Kanade-algoritmus piramisos implementálását használjuk fel. A Lucas–Kanade iteratív optikai folyam számítási módszer megfelelő lokális követési pontosságot eredményez. Az algoritmust pszeudókód formájában összegezve mutatjuk be a [7] és [12] alapján.

Cél: az I kép egy u pontjához találjuk meg a neki megfelelő v helyzetű pontot a J képen.

Készítsük el a I és J képpiramisos reprezentációit (az L felső index a piramis szintjére utal):

 és . (3)

Inicializáljuk a piramisokat a következőkkel:

. (4)

Futtassunk ciklust a legmagasabb szinttől (L = Lm) a legalacsonyabbig (L = 0):

Az **u** pont helye az *IL* képen: **u**L = [*ux  uy*]*T* = **u**/2*L*.

Közelítsük *IL x* szerinti deriváltját a következő differenciával:

*Ix*(*x,* *y*) = (*IL*(*x* + 1, *y*) – *IL*(*x* – 1, *y*))/2, (5)

majd közelítsük *IL* *y* szerinti deriváltját a következő differenciával:

*Iy*(*x,* *y*) = (*IL*(*x,* *y* + 1) – *IL*(*x,* *y* – 1))/2. (6)

A térbeli gradiens mátrixból képezzük a **G** kettős összeget:

. (7)

Inicializáljuk az iteratív Lucas–Kanade-módszer szerint:  = [0 0]*T* .

Iteráljuk a következőket, *k* = 1-től *K* határig (vagy amíg || || < pontossági küszöb):

A képkülönbség:

. (8)

A képeltérés vektora:

. (9)

A Lucas–Kanade optikai folyam:

. (10)

A következő iterációhoz az előkészítés:

 (11)

a *k*-ra vonatkozó ciklus itt lezárul*.*

Az adott *L* szint végső optikai folyama:

. (12)

A következő (*L* – 1) szinthez az érték:

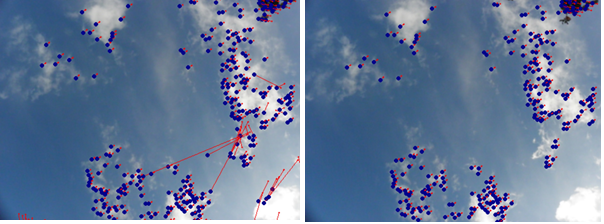
 (13)

Az *L* piramisszintekre vonatkozó ciklus itt fejeződik be.

A végső optikai folyam vektor:

**d** = **g**0 + **d**0. (14)

Tehát az *I* kép **u** pontjának megfelelő pont helyzete a *J* képen a



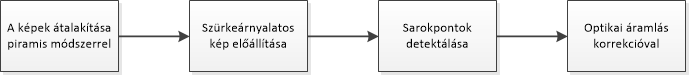
15. ábra – Az optikai áramlás eredménye korrekciók nélkül (balra), majd megfelelő korrigálások elvégzése után (jobbra).

**v** = **u** + **d** (15)

vektorral jellemezhető.

Az algoritmus eredményeit szemlélteti a 15. ábra. Láthatjuk, hogy a kapott eredmény még korántsem mondható megfelelőnek, így bizonyos utófeldolgozásokat kell alkalmaznunk. Ki kell választanunk a számunkra megfelelő összepárosításokat, a téveseket pedig eldobjuk. Azokat a vektorokat, amik a kép méretén túl mutatnak nem szabad detektálni. Amelyek pedig a környezetében lévő többi vektor átlagtól irreálisan eltérnek, szintén el kell vetnünk.

Tehát összefoglalva a szélirány meghatározásához alapvetően két lépésre van szűkségünk: a sarokpontok megtalálására és ezek megfeleltetésére, és az elmozdulás vektorok meghatározására. Azonban ez önmagában nem elég. A pontos eredmények eléréséhez szűkségünk volt előfeldolgozásra, a nem megfelelő eredmények korrigálására (lásd 15. ábra).



16. ábra – A szélirány meghatározásának a lépései.

|  |  |  |
| --- | --- | --- |
| **Vizsgált sorozat időpillanatai** | **Detektált vektorok hibaaránya** | **Elfogadható?** |
| Eredmeny2seb_5 | 1 % | Igen |
| Eredmeny6 | 5% | Igen |
| Eredmeny7 | 3 % | Igen |
| C:\Users\Ádám\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Eredmeny8.jpg | 1 % | Igen |
| C:\Users\Ádám\AppData\Local\Microsoft\Windows\INetCache\Content.Word\OptikaiAramlasKorrekcioval.jpg | 2 % | Igen |
| RosszEredmeny2 | 50 % | Nem |

5. táblázat – Az 1000 db képet tartalmazo sorozat bizonyos időpillanataiban vett eredményeink. Látható, hogy a felhőket más-más forma és szín jellemzi, így az algoritmusunk hatékonysága is változó.

## 3.5. Eredmények kiértékelése

Az OMSZ–tól kapott képsorozaton (1000 db kép) a tesztek többsége elfogadhatónak mondható. Azonban rendelkezünk 1-2 olyan sorozatfelvétellel, amin nem elfogadható a kapott eredmény, így a rendszer további pontosításra szorul. Ennek az az oka, hogy sorozatfelvétel időpontjában a felhők olyan formát és színt öltöttek, amely miatt az algoritmusunk nem tudott megfelelő eredményt produkálni (lásd 5. táblázat).

# 4. Esőzés kezdetének és befejeződésének megállapítása

## 4.1. Cél meghatározása

A meteorológiai állomások nagy gondot fordítanak a rossz időjárási viszonyok előrejelzésére. Ennek óriási jelentősége van a reptereken, a tengeri és a szárazföldi közlekedésben is. Ha egy adott helyen felismerhető lenne egy gépi látó rendszer segítségével, hogy száraz vagy csapadékos az idő, akkor helyettesíthetőek lennének bizonyos speciális meteorológiai eszközök. Megfelelően gyors sűrűséggel készített fényképek mindig a legfrissebb adatokat szolgáltatnák a meteorológiának, vagy egy intézménynek, amely a rendszerünket használja.

Célunk tehát képfeldolgozással lehető legpontosabban meghatározni, hogy épp esik-e az adott időpillanatban, és ebből kikövetkeztetni, hogy mikor kezdett esni, és mikor fejeződött be. Ám esetünkben kényszerfeltételek adottak. Az OMSZ által használt képszolgáltató eszköz egy üvegbúra alatt helyezkedik el, melyet kihasználhatunk, ugyanis az esőcseppek bizonyos ideig megmaradnak az üveg felszínén. Ez önmagában jó hír, ám csak 10 percenként készül három fényképet tartalmazó sorozatkép, emiatt a pontos időpontokat csak megbecsülni tudjuk. További nehézségeket okoz az, hogy a kamerán beállított fókusz végtelenre van állítva, így az elkészült képeken élesen kivehetőek az esőcseppek illetve a háttér is, mely lehet egy torony, egy városrész, attól függően, hogy hol helyezték el a kamerát. Ahhoz, hogy kialakítsuk a lehető legoptimálisabb algoritmust, ismernünk kell, hogy milyen tulajdonságokkal rendelkezik egy esőcsepp, mely egy üveglapon – esetünkben egy burán – landol.

## 4.2. Esőcseppek fizikai jellemzői

Az esőcseppek becsapódáskor egy gyors torzuláson mennek keresztül, mely jelenség gyakran valamilyen szintű rezgésre utal. Mivel a rezgés számunkra nem szignifikáns, ezért az esőcseppet vehetjük egy fix alakzatnak. Az, hogy ez hogyan néz ki, függ a méretétől. Egy kisebb csepp inkább gömbölyded formát ölt, míg a méret növekedésével egyre inkább egy összenyomott szferoid felé tendál. [13]

Számunkra még ennél is fontosabb jellemzők a következők [14]:

* Azon esőcseppek, melyekre a kamera ráfókuszált a kép készítésekor igen jól elszeparálható objektumokként jelennek meg a háttérhez viszonyítva. Ezzel ellentétben a nem fókuszban lévő cseppek nehezen elkülöníthetőek.
* A cseppek átlátszóak.
* Világosabbak a háttértől, mely az égbolt fényerejének köszönhető. (lásd 17. ábra)
* A fókusz csökkenésével a cseppek mérete egyre nagyobbnak tűnik és kevésbé detektálható az erősen elmosódott hatás miatt.



. ábra - Megfigyelhetőek az esőcseppek jellemzői, többek között az, hogy világosabbak, mint a háttér. Észrevehető, hogy éldetektálással azonban

Továbbá megfigyelhető még az üveg felszínén, hogy a gömbölyded formája miatt a háttérben látható elemeket tükrözve láthatjuk, feltéve, ha a csepp a fókuszban van. Esetünkben a kamerák végtelenre állított fókusza miatt egyaránt figyelhetőek meg éles és elmosódott cseppek - ez utóbbiak főleg a világos területeken -, így minden szempontot figyelembe kell vennünk detektáláskor. Éldetektáló algoritmusunk remekül alkalmazható azon esetekre, amikor a cseppek élesen elkülönülnek a háttértól, ugyanis részben, vagy egészben a szélein lévő pixelek intenzitás értékei különböznek a közvetlenül mellette lévő háttér intenzitásértékeitől. E metódusok akár a teljes esőcsepp körvonalát is felismernék, azonban nem mindig van ilyen szerencsénk.

## 4.3. Lehetőségek eső detektálására

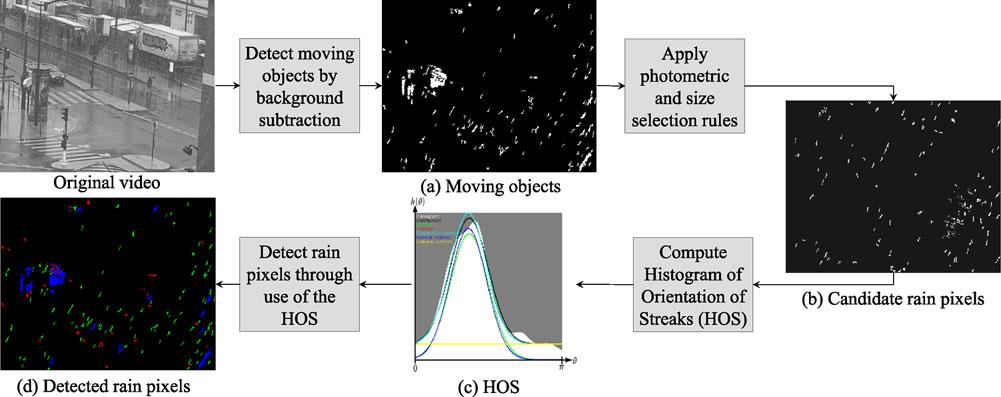
### 4.3.1. Esővonalak detektálása hisztogram vizsgálattal mozgókép sorozaton

Létezik olyan megközelítése a problémának, amely egy videó képeinek a feldolgozásával kísérli meg jelen esetben az eső vonalak felismerését. Az egyes elkészült képeken nem keresünk esőcseppeket, ugyanis a cseppek zuhanási sebessége miatt vonalakként látszanak. Ha két egymást követő képet kivonunk egymásból, akkor többek között ezek a vonalak is látszódni fognak, mint különbség. Következő feladatunk kiszórni azokat az objektumokat, melyek nem az esővonalak által generált különbséget eredményezik. [15]

E módszer szerint először szükségünk van két kiválasztó szabály alkalmazására, egyik a fényerősségen alapszik, a második pedig az objektumok méretén. Az egyes területeken az előtér intenzitásértékeit kivonjuk a háttérmodell intenzitásértékeiből a (21)-es képlet szerint. Ha az eredmény egy adott küszöbérték fölött van, akkor feltehetőleg egy esővonalról van szó. Továbbá a méreteiben kiugró objektumokról feltételezhetjük, hogy nem esőcseppek, hanem például egy mozgó gépjármű, vagy egy fa, amelyet a szél mozgatja. [15]

*ΔI = IFG −IBG ≥ c* (21)

Végül az objektumok orientációi alapján eldöntjük, hogy mely orientáció az, amely legjobban jellemzi az esővonalakat. Ehhez az egyes irányokból hisztogramot számolunk, majd ez alapján magabiztosan megállapíthatóak a keresett vonalak (lásd 18. ábra). [15]



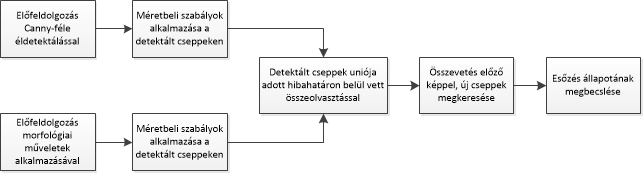
18. ábra - Eső detektálásának folyamata mozgóképen: (a) mozgó elemek detektálása háttér kivonásával; (b) fényerősségi és méretbeli szabályok alkalmazásával a potenciális területek kiválasztása; (c) a vonalak irányaiból hisztogramot számolunk;(d) esőpixelek felismerése a hisztogram felhasználásával.

Sajnos esetünkben a mozgókép készítése nem valósítható meg, így az esővonalak detektálásának nincs értelme. A legrövidebb időn belül elkészült képek között is eleredhet vagy megállhat az eső, így a két kép között már valamilyen tartósabb jellemzőket kell keresnünk, erre pont megfelelőek a búrán megmaradt esőcseppek.

### 4.3.2. Esőcseppek felismerése gépjármű szélvédőjén

Egy másik megközelítés az, ha az esőcseppeket keressük az adott képeken. Mivel egy átlagos kamera, webkamera nem képes olyan rövid záridővel fényképet készíteni, melyen a levegőben kivehetőek lennének a cseppek, így egy olyan helyet szükséges lefényképezni, ahol már becsapódott. Viszont ebben az esetben is különféle eshetőségekkel is foglalkoznunk kell, ilyen például az, amikor a kamera nem fókuszál rá a felületre – például üveglapra –, amelyre ráesett a csepp. Egy igen jó példa erre az, amikor gépjármű vezetése közben esős időben a szélvédőn megfigyelhetjük a vízcseppeket.

Ha egy kamerát helyeznénk el a szélvédő belső oldalán, amely mozgókép felvételére képes, akkor az elkészült képeket vizsgálva azt tapasztalnánk, hogy bár a háttér folyamatosan mozgásban van – feltéve, ha az autó is mozgásban van –, de néhány képsorozaton keresztül az esőcsepp mozdulatlan marad. Ezeket a jellemzőket felhasználva könnyedén felismerhetőek az egyes elmosódott foltok, melyek a nem fókuszban lévő esőcseppek miatt keletkeztek. [14], [16]



. ábra - Esődetektálás teljes folyamata

## 4.4. Esőcseppek detektálásának folyamata üvegbúrán

### 4.4.1. Előfeldolgozás a jól elkülöníthető cseppek detektálásához

Rendszerünk egyik legnagyobb kihívása, hogy a búrára becsapódó cseppeket úgy detektáljuk, hogy eközben minél kevesebb olyan objektumot ismerjünk fel, amely nem releváns. Valamilyen szegmentációs algoritmusra van szükség, mellyel a különböző pacákat kiválogatjuk. A kiválogatás után pedig meg kell határozni, hogy melyek a keresett objektumok. Ám előbb az input képnek egy előfeldolgozási folyamaton kell átesnie, aminek eredményeképp a szegmentáció eredményesebb lesz.

A kamerából érkező kép az eredeti felbontásában és képminőségében érkezik. Nincs szükségünk feltétlenül a teljes felbontásra, ráadásul a teljes detektálási folyamatot is lelassítaná. Így először egy 0,5 megapixeles képpé alakítjuk át, így a felbontása 800×600 pixel lesz. Mivel a színeknek nincs szerepe a folyamatban, így még azzal is gyorsíthatjuk az algoritmust, ha szürkeárnyalatos képet készítünk.

Következő feladatunk a zajszűrés. A háttér lehet bármi, így a legfontosabb feladatunk a zavaró pixeleket egy zajszűrő algoritmussal kiszűrni, erre legalkalmasabb a medián szűrő. Az így létrejött képen még továbbra is nehezen elkülöníthetőek az élek. Ahhoz, hogy a szegmentációs algoritmus ne eredményezzen óriási mértékű hibás elemeket, el kell érnünk, hogy a háttér elmosódjon anélkül, hogy az élek még kivehetetlenebbek legyenek. Így el kell végeznünk egy éldetektálást, majd az élek mentén élesíteni, a háttért pedig simítani kell. Erre alkalmas algoritmus az adaptív simítás, mely során a gradiensek mértékétől függ a simító eljárás során alkalmazott ablak mérete. [17]



20. ábra - Éldetektálással végzett esőcsepp detektálás előfeldolgozásának az eredménye.

Nincs más hátra, mint a kiemelt éleket egy megfelelő módszerrel kiválogatni, majd erősségük szerint osztályozni őket. A Canny-féle éldetektálás megfelelőnek tűnik, paraméterektől függően képes az az esőcseppek nagy részét, és még ha a fizikai jellemzőknél leírt okok miatt nem is a teljes kontúrt, de nagy részét megfelelően megtalálja (lásd 20. ábra).

### 4.4.2. Előfeldolgozás morfológiai műveletek alkalmazásával

Ahogy az esőcseppek fizikai jellemzőinél is kiemeltük, az OMSZ által biztosított kamerák végtelen fókuszra vannak állítva. E miatt és az időjárási viszonyok miatt sajnos előfordulnak olyan esetek, amikor az esőcsepp csak egy elmosódott paca a képen, így a fent ismertetett előfeldolgozási algoritmus által előkészített képen nem kivehető az összes esőcsepp. Legjobb megoldás, ha egy hibrid algoritmust használunk, mely részben elvégzi a fent említett képen a blob-detektálást, ám felkészítjük a rendszerünk egy új algoritmus lefuttatására, amely az elmosódott, nehezen kivehető vízcseppekre van kihegyezve.

A képen, a fentebb leírtak szerint valamilyen tömörítést kellene legelőször végrehajtani, majd szürkeárnyalatosítani, végül zajmentesítés szükséges. Tapasztalataink szerint az adaptív simítás [17] itt is jó hatásfokot eredményez, így érdemes ezt is lefuttatni a képeken.

A folyamat itt válik el az előbb ismertetett algoritmustól, ugyanis ezúttal morfológiai műveletekkel szeretnénk kiemelni a foltokat. E módszer szerint először egy 20 pixel átmérőjű ablakkal kell erodálni az előállt simított képet. Ezzel elértük, hogy a sötét és világos területek - legtöbb esetben a horizont alatti és fölötti rész - elkülönülnek egymástól. Majd egy morfológiai szűréssel, ahol az erodált kép lesz a maszk, állítsuk helyre a simított képet. Ismételjük meg ezt a folyamatot ám ezúttal az invertált szűrt képen (lásd 21. ábra). [18]

### 4.4.3. Esőcseppek felismerése hibrid szegmentáló algoritmussal

A folyamat következő lépcsőfoka a szegmentálás. Jelenleg létrejött két képünk, melyeken valamilyen szegmentálást kellene végrehajtanunk. Az első képen már csak annyi dolgunk van, hogy megkeressük a jól kiemelt pacákat, szűrjük őket méretbeli szabályok alkalmazásával, majd rögzítsük a koordinátáit és méreteit a későbbi felhasználás céljából. A második képen viszont további műveletekre van még szükség.



21. ábra - Morfológiai műveletekkel végzett esőcsepp detektálás előfeldolgozásának az eredménye.

Az első és második képen is blob-detektálást kell elvégeznünk. Ez rengeteg értékkel tér vissza, melyek között sajnos háttérzaj is található. Ezen kívül észrevehető, hogy az esőcseppek több részletben is megtalálhatóak. Ennek oka az esőcseppek tulajdonságaiból adódik. Ha a háttérben erőteljes kontraszt van a horizont fölött és alatt, akkor a cseppben megfigyelhető fordított virtuális kép is hasonló jellemzőkkel bír majd, ennek megfelelően előfordulhatnak olyan pixelek a szélein, ahol az intenzitás értékek majdnem vagy teljesen megegyeznek a háttér szomszédos pixeleinek intenzitásértékeivel, így ott az éldetektáló algoritmus nem eredményez szignifikáns eltérést.

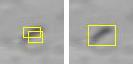
Szükséges szűrést végrehajtani a detektált területeken, először méretbeli szabályok alkalmazásával. Ezek a különböző foltok méretére értendőek, melyet az egyszerűség kedvéért egy köré húzott téglalappal is jellemezhetünk. Mivel egy-egy esőcsepp a fentebb leírt okok miatt több foltként is detektálásra került, így első dolgunk valamilyen algoritmussal ezeket összeilleszteni.

Tapasztalataink szerint, ha bármely két téglalapot legalább 40%-ban fed egy másik téglalap, akkor ezek a későbbiekben már egy objektumnak tekinthetők, további számítások során pedig azzal a legkisebb téglalappal dolgozunk, amelybe belefér a másik kettő (lásd 22. ábra). A további pontosítás érdekében a szűrés előtt hasznosnak találtuk minden egyes téglalap méretét 150%-kal megnövelni úgy, hogy középpontjuk ugyanazon a koordinátán maradjon, így az intenzitásokból fakadó problémák kiküszöbölhetőek.

Végezetül szükség lesz a két kép eredményeinek összevetésére. A megtalált elemek unióját véve, majd a fentebb leírt méretbeli szabályok alkalmazásával szűrt objektumok lesznek az új hibrid algoritmusunk detektálásának eredménye. Így elértük azt, hogy ha egy csepp az éldetektálós módszerrel csak félig lett kiemelve, vagy épp két esőcseppként lett számba véve, akkor végül mind koordináták alapján, mint pedig a darabszámok alapján is egy pontosabb végeredmény születik.

## 4.5. Kezdeti és befejeződési időpont becslése

Az eredeti célunk nem az esőcseppek pontos számának a megállapítása, hanem a kezdeti és befejeződési időpont megbecslése. Ehhez azonban két egymást követő képet kell összehasonlítani. Minél közelebb készültek egymáshoz képest időben, annál pontosabb becslést tudunk adni. Az eldöntés alapja pedig a két képen vett esőcsepp detektálások eredményeinek összevetése.



22. ábra - A szegmentálás során egy esőcseppen akár több folt is detektálásra kerül (balra), ám méretbeli szabályok alkalmazásával ezek hatékonyan egyesíthetők, így pontosítva a cseppek számának a megbecslését (jobbra).

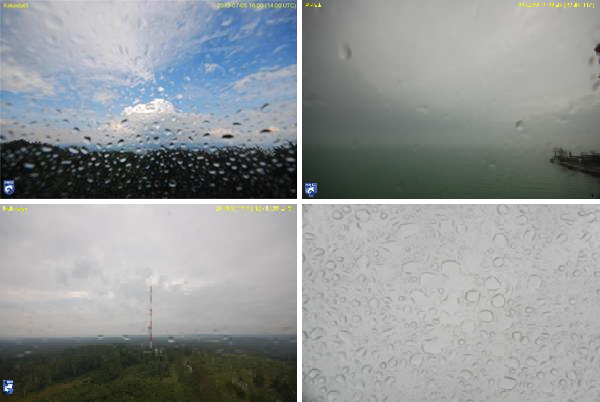
Először meg kell keresnünk azokat a cseppeket, amelyek ezúttal nem egyforma koordinátán vannak. Kiszórjuk azokat a téglalapokat, amelyek legalább 80%-ban lefedik bármely az előző képen detektált területet. A megmaradó esőcseppeket pedig tekinthetjük új, azaz az előző időpillanat óta létrejött cseppnek. Ennek előnye az is, hogy ezzel kizártuk annak a lehetőségét, hogy egy piszok, vagy egyéb hibásan felismert háttérből származó objektum esőcseppként legyen felismerve, ugyanis ezek nagy valószínűséggel már az előző képek során is ott voltak és detektálásra kerültek.

Számunkra elegendő információt ad az, ha tudjuk, hogy az előző képhez képest arányaiban több, vagy kevesebb csepp van. Ha azt tapasztaljuk, hogy több csepp van, mint az időben korábbi képen, akkor feltételezhetjük, hogy azóta esett eső. Ha a cseppek száma jóval kevesebb, mint az első vagy a másodikon detektált összes esőcsepp, akkor feltételezhető, hogy már száradás van folyamatban, így a két kép között bizonyára abbamaradt az esőzés.

# 4.6. Elért eredmények

Az OMSZ számunkra biztosított képgyűjteményben 4 olyan képsorozat található, amelyeken van értelme esődetektálást tesztelni (lásd 23. ábra). Egy-egy sorozat a reggeli óráktól kezdve sötétedésig negyedóránként lőtt fotókat tartalmaz. Ezekből egyiken a kamera fölfelé nézett. Az elért eredmények azonban megoszlanak.

A legnagyobb sikert a Kékestetőn elhelyezett horizontot figyelő kamera képein értük el. Háttér szegényes, nagy kontraszt van az égbolt és a föld között, így a megjelenő esőcseppek nagy részét szegmentálni tudtuk, így meg tudtuk becsülni az időpontokat 15 perces hibahatárral. A siófoki és a kab-hegyi sorozaton a detektált esőcseppek száma miatt – mely adódik az alapból kevés becsapódó és nehezen szegmentálható cseppekből – általában késve jelezte a rendszer, hogy elállt az eső. Végül a fölfelé néző kamera esetében a cseppeket könnyedén felismertük, ám a ritkán készített képek miatt, a cseppek helyzete mindig változott. Emiatt bár a kezdeti időpont általában pontos volt, a befejeződést szintén késve jegyezte fel a rendszer.



. ábra – Algoritmusunk tesztelése különböző helyszíneken. Sorrendben balról jobbra, fentről lefelé: Kékestető, Siófok, Kab-hegy, ismeretlen helyszínen fölfelé néző kamera.

# 5. Összegzés

## ****5.1. Elért eredmények****

A programunkban megvalósított modulok külön-külön is hasznosak, azonban együttes alkalmazásukkal majdnem teljes jelentést készíthetünk egy adott terület időjárására vonatkozóan. Az Országos Meteorológiai Szolgálat rendkívül hasznosnak találta az alkalmazást, és amennyiben megfelelő pontosságot tudunk elérni, örömmel alkalmaznák is az észlelők segítése érdekében.

Felhők típusának meghatározása, és borultság vizsgálata esetén a program 87.06%-os pontossággal detektált 491 képet vizsgálva. Eső detektálásnál az algoritmus nem csak megtalálja az esőcseppeket, de a hozzájuk hasonló statikus foltok (pl. karcolás, kosz) kiszűrésére is képes. Felhők mozgásirányát vizsgálva az algoritmus a legtöbb esetben képes helyes eredményt adni. Ezen kívül a program meg tudja állapítani, hogy a fényviszonyok lehetővé teszik-e a tesztelést, valamint felfelé néző kamerán lévő esőcseppek detektálására, és csapadék melletti felhő felismerésre is sor került. Mivel a program nem azért készült, hogy az észlelők helyett dolgozzon, hanem hogy munkájuk segítse és pontosítsa a becsléseiket, így a célnak sikerült eleget tennünk.

## ****5.2. Továbbfejlesztési lehetőségek****

Mivel bizonyos kényszerfeltételek adottak voltak, így legnagyobb célunk az volt, hogy a képfeldolgozó algoritmusainkat optimalizáljuk arra, hogy akár több perc is eltelhet két fénykép elkészülése között. Két kép közötti összehasonlítás eredménye akkor a legpontosabb – főként a magassági szélirány és az esőzés detektálása esetében –, ha másodperceken belül több fotó is készülne a környezetről. Erre azonban egy mozgókép felvételére alkalmas eszközt kellene használni.

Magyarországon a rendszer egyedülállónak számít, melyet az Országos Meteorológiai Szolgálat is megerősített. Mivel az észlelők munkáját nagyban elősegítjük ezzel az automatizált rendszerrel, így mindenképpen egy hasznos projektkísérletről van szó. Amellett, hogy részben helyettesíti a munkát, sok esetben akár pontosabb értékekkel is szolgálhatunk, főként a borultság vizsgálatának az esetében, ugyanis ez esetben teljes mértékig az észlelő szemére van bízva az érték meghatározása.

Második fontos szempont esetünkben az, hogy egy olcsó fényképezőgép is képes hasznos nyújt számunkra, tehát nincs szükség költséges eszközökre, például műholdas rendszerekre. Ez a megoldás a világ bármely területén költséghatékonynak számít, így más szervezetek, cégek is fel tudják használni ezt a technológiát.

# Irodalomjegyzék

[1] *Gépi látó rendszerünk fejlesztésének a támogatására elkészült írásos specifikáció és képgyűjtemény*. Budapest: Országos Meteorológiai Szolgálat, 2013.

[2] *Cloud types for observers, Reading the Sky*. MET Office, 2006.

[3] F. Samopa and A. Asano, “Hybrid Image Thresholding Method using Edge Detection,” *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 9, no. 4, pp. 292–299.

[4] M. P. Souza-Echer, E. B. Pereira, L. S. Bins, and M. A. R. Andrade, “A Simple Method for the Assessment of the Cloud Cover State in High-Latitude Regions by a Ground-Based Digital Camera,” *Journal of Atmospheric and Oceanic Technology*, vol. 23, no. 3. pp. 437–447, 2006.

[5] R. A. Haddad and A. N. Akansu, “A class of fast Gaussian binomial filters for speech and image processing,” *IEEE Trans. Signal Process.*, vol. 39, no. 3, 1991.

[6] M. S. Nixon and A. S. Aguado, “Feature Extraction and Image Processing,” *Acad. Press*, 2008.

[7] B. K. Choudhary, N. K. Sinha, and P. Shanker, “Pyramid Method in Image Processing,” *J. Inf. Syst. Commun.*, vol. 3, pp. 269–273, 2012.

[8] J. Bouguet, “Pyramidal implementation of the affine lucas kanade feature tracker—description of the algorithm,” *pages.slc.edu*, 2001.

[9] E. Trucco and A. Verri, “Introductory Techniques for 3-D Computer Vision,” *Prentice Hall*, 1998.

[10] J. M. R. D. G. Sainarayanan, “Harris Operator Corner Detection using Sliding Window Method,” *Int. J. Comput. Appl.*, vol. 22, no. 1.

[11] D. Fleet and Y. Weiss, “Optical Flow Estimation,” in *Mathematical models for Computer Vision: The Handbook*, Springer, 2005, pp. 239–257.

[12] Z. Vámossy, Á. TóthP, and Hirschberg, “PAL Based Localization Using Pyramidal Lucas-Kanade Feature Tracker,” *IEEE Proc. 2nd Serbian-Hungarian Jt. Symp. Intell. Syst.*, 2004.

[13] K. Garg and S. K. Nayar, “Vision and Rain,” *International Journal of Computer Vision*, vol. 75, no. 1. pp. 3–27, 2007.

[14] F. Nashashibi, R. de Charrette, and A. Lia, “Detection of unfocused raindrops on a windscreen using low level image processing,” *2010 11th Int. Conf. Control Autom. Robot. Vis.*, pp. 1410–1415, 2010.

[15] J. Bossu, N. Hautière, and J.-P. Tarel, “Rain or Snow Detection in Image Sequences Through Use of a Histogram of Orientation of Streaks,” *International Journal of Computer Vision*, vol. 93, no. 3. pp. 348–367, 2011.

[16] H. Kurihata, T. Takahashi, I. Ide, Y. Mekada, H. Murase, Y. Tamatsu, and T. Miyahara, “Rainy weather recognition from in-vehicle camera images for driver assistance,” *IEEE Proceedings. Intell. Veh. Symp. 2005.*, 2005.

[17] C. Kervrann, “An Adaptive Window Approach for Image Smoothing and Structures Preserving,” *Processing*, pp. 132–144, 2004.

[18] A. Cord and G. Nicolas, “Detecting Unfocused Raindrops In-Vehicle Multipurpose Cameras,” *IEEE Robot. Autom. Mag.*, vol. 21, no. 1, pp. 49–56, 2014.