



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. Vámosy Zoltán
egyetemi docens

2014

TARTALOMJEGYZÉK

Tartalomjegyzék	2
1 Rendszer célja	4
1.1 Észlelők munkájának támogatása	4
1.2 Működtetés kényszerfeltételek mellett	5
2 Borultság vizsgálata és felhőtípusok osztályozása.....	7
2.1 Cél meghatározása	7
2.2 Felhők fajtái és jellemzői.....	8
2.3 Probléma elemzése	12
2.4 Hasonló rendszerek.....	13
2.5 Borultság vizsgálata hibrid küszöböléses algoritmussal.....	13
2.6 Borultság vizsgálata szaturáció méréssel.....	16
2.7 Borultság vizsgálat megvalósítása	19
2.8 Felhők osztályozása képfeldolgozással	23
2.9 Speciális esetek kezelése	24
2.10 Eredmények értékelése	26
3 Magassági szélirány meghatározása	29
3.1 Cél meghatározása	29
3.2 Probléma elemzése	30
3.3 Megvalósítás	30
3.4 Eredmények értékelése	33
4 Esőzés kezdetének és befejeződésének megállapítása.....	35
4.1 Cél meghatározása	35
4.2 Esőcseppek fizikai jellemzői	35
4.3 Az esődetektálás lehetőségei	36
4.3.1 Esővonalak detektálása hisztogram vizsgálattal mozgókép sorozaton	36
4.3.2 Esőcseppek felismerése gépjármű szélvédőjén.....	37
4.4 Esőcseppek detektálásának folyamata üvegbúrán	38
4.4.1 Előfeldolgozás a jól elkülöníthető cseppek detektálásához.....	38

4.4.2	Előfeldolgozás morfológiai műveletek alkalmazásával	39
4.4.3	Esőcseppek felismerése hibrid szegmentáló algoritmussal	40
4.5	Kezdeti és befejeződési időpont becslése	41
4.6	Elért eredmények	41
5	Továbbfejlesztési lehetőségek.....	44
6	Összegzés	45
7	Irodalomjegyzék.....	47
8	Ábrajegyzék	49
9	Mellékletek.....	52
10	A programrendszer bemutatása.....	57
10.1	A rendszer moduljai.....	57
10.2	Fő modul	58
10.2.1	Felhasználói felület.....	58
10.3	Interfészek.....	59
10.3.1	Image Process Service interfész	60
10.3.2	Subprocess Service interfész	61
10.4	Kisegítő osztályok.....	62
10.5	Felhő detektáló modul	62
10.6	Magassági szélirány detektálása modul	65
10.7	Esőzés detektálása modul	66
10.8	Fejlesztői verzió	69

1 RENDSZER CÉLJA

1.1 ÉSZLELŐK MUNKÁJÁNAK TÁMOGATÁSA

Az Országos Meteorológiai Szolgálat (továbbiakban OMSZ) észlelő állomásain – például Siófokon – manapság az időjárási jelenségek jelenetős részének mérése még becsléssel, az észlelők által történik. Ezek közül néhány megfigyelést 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 számos nagy felbontású képet szolgáltatott nekünk, amelyek nagyban segítettek a munkánkat. Emellett sikerült szert tennünk egy, a kihelyezett kamerákkal megegyező felépítésű gépre, búrával és fűtő rendszerrel együtt, amivel a képek mennyiségét tovább növelhettük, és teljes mértékben reprodukálni tudtuk az éles környezetet. Ilyen módon esőben készült képekhez is sikerült hozzájutni. Mindezek mellé az észlelések során készített hivatalos dokumentum is elérhetővé vált számunkra, ami lehetővé tette a tesztek pontos elvégzését, és így elősegítette a statisztikák, grafikonok elkészítését. 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 többnyire 1 órás időközökkel a nap 24 órájában történnek. A felhőzet vizsgálatát az észlelők teljesen a saját becsléseikre hagyatkozva végzik. Egyes szakemberek már több mint 20 éves tapasztalattal rendelkeznek ezen a téren, de természetesen találunk új észlelőket is. A becslések így eltérők, és pontatlanok lehetnek. 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 az észlelőket műszerek támogatják az állomásokon, így ezeket a meteorológiai adatokat pontosan tudják feljegyezni. Ezzel szemben a kamerák mellé nincsenek ilyen műszerek kihelyezve, így ott a csapadék mennyiségét illetve jelenlétét nem mérik, míg a már jelen lévő kamerák segítségével ez lehetséges lenne. Megállapítható, hogy igazán pontos eredményeket az informatikai rendszerek és az emberi becslések együttes alkalmazásával kaphatunk.

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. Megvalósítás terén ezeket a modulokat külön készítjük el, de

egy közös interfész segítségével egy felületben egyesítjük, és lehetővé tesszük, hogy egymás eredményeit is felhasználhassák a pontosabb eredmények érdekében.

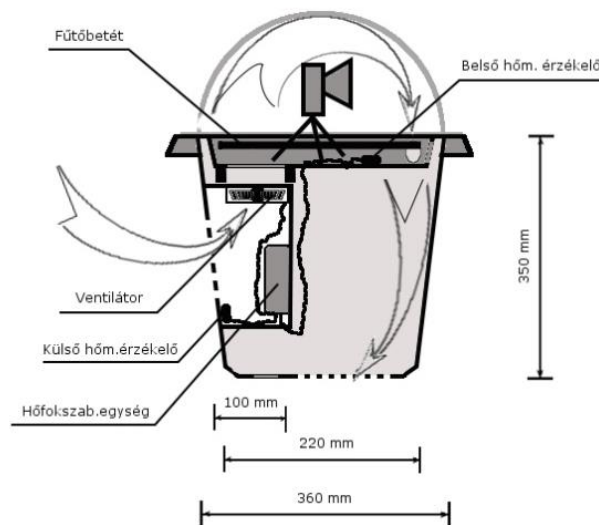
1.2 MŰKÖDTETÉS KÉNYSZERFELTÉTELEK MELLETT

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 (lásd 1.1. ábra) vannak használatban, nagy látószögű objektívvel. A rendszer fejlesztőinek választása azért erre a készülékre esett, mivel számítógépre csatlakoztatva különböző utasításokkal vezérelhető, így egyszerűen megoldható a képek ütemezett készítése, továbbítása más rendszerek számára. Hátránya viszont, hogy tükörreflexes, amiből adódóan gyorsabban hibásodik meg a gyakori fényképezéstől. A kamerákat burok veszi körül, ami védi a viharoktól, portól, és egyedi fűtésrendszerrel rendelkezik, így nem párasodik, illetve télen-nyáron egyaránt használható (lásd 1.2. ábra). Vezérlésüket egy speciális router végzi, amire operációs rendszer van telepítve. A kamerák jelenleg 10 perces időközökkel készítenek 1–1 képet, viszont a programunk egyes funkcióihoz (magassági szél irányának meghatározása, esőzés figyelése) egymás után gyorsan elkészített képek szükségesek. Sajnos az OMSZ nem biztosítja számunkra azt, hogy a fényképezőgép minden másodpercben fényképezzen, mivel a gépek tükörreflexesek, és így ezzel a kamerák mechanikájának 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.



1.1. ábra - A rendszer fejlesztése során használt NIKON D50 fényképezőgép, melyhez hasonlókat jelenleg az ország több mint 9 pontján készítenek rendszeresen képeket az OMSZ számára [1].

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 maradhatnak a kamera előtt. 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. Az OMSZ szakemberei már hosszú ideje dolgoznak a jelenlegi kamerákat irányító és védő rendszer továbbfejlesztésén, olyan módon, hogy a gépet tartó elem számítógéppel vezérelve minden irányban, fokokkal megadott pontossággal elforgatható legyen. Ez régebben egyszer már meg lett valósítva, azonban a vezetékek hamar elkoptak a súrlódástól. Jelenleg ennek kiküszöbölése, és a már meglévő rendszerbe való beépíthetőség a fő cél. Ennek segítségével programunk használata nem igényelne két kamerát, mivel forgatás segítségével könnyen beállíthatjuk az irányt a nekünk megfelelőbe.



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

2 BORULTSÁG VIZSGÁLATA ÉS FELHŐTÍPUSOK OSZTÁLYOZÁSA

2.1 CÉL MEGHATÁROZÁSA

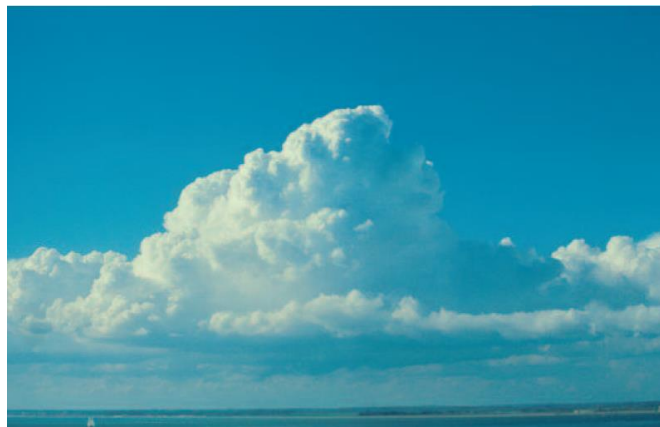
A meteorológiai észlelők az ég borultságát is vizsgálják. Az elnevezés ellenére ez nem a felhőzet színét jelenti, hanem azt, hogy a belátott területet az égen mekkora mértékben fedi felhő. Ezt az értéket úgynevezett oktában mérik, ami egy 0-tól 8-ig tartó osztályozási rendszert jelent. Ha az érték 8, akkor az eget teljesen elfedik a felhők, míg 0-nál teljesen felhőtlen. Azonban adódnak bizonyos speciális esetek. Amennyiben az értéket 8 oktának becsültük, de valahol van kis lyuk a felhőzetben, az értéket 7 oktának kell venni. Ugyan így 0 oktánál, ha egy egészen kis felhő van az égen, már 1 okta az értéke. Ezen kívül fontos figyelembe vennünk, hogy Cumulus típusú felhő (később részletezve) nem lehet 8 okta. Ebben az esetben feltételeznünk kell, hogy vannak kis lyukak, csak mi nem látjuk őket, és az értéket 7 oktának kell becsülni. A mérés jelenleg ránézéses becsléssel zajlik. A hivatalos módszer, hogy a belátott területre képzeletben egy nyolc részből álló rácsot illesztünk, ami alapján megbecsülhető, hogy az összes felhő hány cellát foglal el (lásd 2.1. ábra). Ez az érték adja a borultságot. [1]

Hogy ezt az értéket képeken is megkaphassuk, első lépésként el kell tudnunk különíteni a felhőket az égtől. Ha ezt sikerült elvégeznünk, akkor ez az érték már könnyen kiszámítható a kamerán látható területre vonatkozóan. Tehát az első feladat számunkra a felhők detektálása, égtől való elkülönítése, majd az imént említett feltételek mellett meg kell határozni az ég oktában mért borultságát.



2.1. ábra - Égbolt borultságának meghatározása becsléssel, képzeletbeli rács használatával. A kép alapján becsült érték 3 okta [2].

A felhők típusának megállapítása már egy összetettebb feladat. Ez abból is látható, hogy ehhez is szükséges a felhők elkülönítése az égtől, viszont a detektált felületeket tovább kell vizsgálnunk. Tovább nehezíti a feladatot, hogy 10 alap típusú felhőt különböztetünk meg, melyeket további altípusokra bonthatunk. Míg a borultság vizsgálatban jelenleg a gépek pontosabbak, ebben a témakörben még az észlelők becsülik meg pontosabban a 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, ilyen például a kialakulásának folyamata (lásd 2.2. ábra).



2.2. ábra – Cumulus típusú felhő, melynél jól látható a felfelé való terjeszkedés, mely egy határ után elterül. A folyamat egyetlen kép alapján nem detektálható, felfelé néző kamera esetén a felső rész nem is látható [2].

2.2 FELHŐK FAJTÁI ÉS JELLEMZŐI

A felhőket magasság szerint három rétegbe, és ezeken belül 10 fajta oszthatjuk (lásd 1. melléklet). Ha igazán részletesek akarunk lenni, akkor még ezeket is tovább oszthatjuk úgynevezett változatokra is. Ezek meghatározása az észlelői munkakör legbonyolultabb feladata. A kezdőket általában tapasztalt kollégák segítik a gyakorlati tanulásban. Az alábbi adatokat a MET Office könyvében [2] leírtak alapján, valamint az OMSZ specifikációjából [1] gyűjtöttük össze.

Magasan lévő felhők a Cirrus, Cirrocumulus és a Cirrostratus. Ezekre éghajlatunkon igaz, hogy 4000–13000 méter között helyezkednek el, és kizárólag szilárd részecskéket, jeget tartalmaznak. Nappal mindig fehérek, részben átlátszók lehetnek, csapadék nem származik belőlük.

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 kondenzcsíkja is Cirrus-szá alakul, ha sokáig megmarad (és ez nem baj, nem kell leszűrni a felhők közül). Alfajai a Cirrus uncinus és a Cirrus spissatus.



2.3. ábra – Halo jelenség különböző megnyilvánulásai vékony felhőzet esetén. [2]

A Cirrocumulust a köznyelv "báránfelhő"-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, vékony, fehér 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 (lásd 2.3. ábra).

A középmagasan elhelyezkedő felhők 2500–7000 méter között találhatók. Folyékony és szilárd halmazállapotú részecskéket egyaránt tartalmaznak. Nap közben világos, míg hajnalban sötét színezetűek. Csapadék nem, vagy csak minimális intenzitással származik belőlük. Típusai az Altocumulus, Altostratus és Nimbostratus.

Az Altocumulus egy gomolyos, változatos színű középmagasan elhelyezkedő felhő. Elemei határozottan elkülöníthetők. Nem csak a színe, hanem az alakja is változik. Állhat magányosan, padokban, sávokban, vagy az egész égboltra kiterjedve. Csapadék általában nem származik belőle, kivéve az AC7-et, amiből gyenge csapadék származhat. Változó mennyiségben van jelen az égen, 1–7 októáig. Alfajai az Altocumulus translucidus, Altocumulus opacus, Altocumulus duplicatus, Altocumulus lenticularis, Altocumulus floccus és az Altocumulus castellanus.

Az Altostratus is középmagasan van, és a teljes eget beborítja. Általában egysíkú, szürkés vagy kékes színű, és nem ad csapadékot. Két alfaja van, az Altostratus translucidus és az Altostratus opacus. Előbbit átlátszik a Nap, a másikon nem látszik át, vastag és sötét.

A Nimbostratus egy vastag, egész eget beborító. Alja általában 2500 méter alatt található, de az előírások szerint közép magas szintbe kell sorolni. 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. Egyedi jellemzője, hogy esőn és havon kívül fagyott eső is származhat belőle.

Az alacsonyan elhelyezkedő felhők 2500 méter alatt találhatók. Főként vízcseppeket tartalmaznak, hideg légtömegben időnként szilárdot és folyékonyt vegyesen. Nap közben sötétebbek, éjjel viszont világosabbak, mint a magasabb szinten lévő felhők. A legtöbb csapadék hazánkban ebből a rétegből származik. Fajtái a Cumulus, Cumulonimbus, Stratocumulus és a Stratus.

A Cumulus egy alacsonyan elhelyezkedő felhő. Nem boríthatja teljesen az eget, 7 oktánál nem lehet több. Sűrű, éles körvonala van, függőleges irányban gomolyszerűen fejlődik. Az alja egyenes, de egy habos, fölfelé növekvő tetőrész indul ki belőle. Ha kellően magas, akkor závorszerű eső vagy hó származhat belőle. Többnyire derült égen található, nem érnek össze, de gyakran más felhők alatt vannak, vagy más típusokba olvadva. Alfajai a Cumulus humilis, Cumulus mediocris, Cumulus congestus és a Cumulus fractus.

A Cumulonimbus a Cumulus egy továbbfejlődött változata. Akkor nevezzük így a felhőt, ha a Cumulus magassága átlép a közép magas szintre, felhőtoronnyá alakul. 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. Alfajai a Cumulonimbus calvus és a Cumulonimbus capillatus.

A Stratocumulus alacsonyan elhelyezkedő párna alakú felhő. Nagy méretű, gömbölyded elemek alkotják. Előfordulhat sávokba rendezetten, magányosan és az egész égboltra kiterjedve is. Sötét foltok vannak rajta, alja sötétszürke, 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. Általában sötétszürke, egysíkú, egyszínű, és szitálást eredményezhet. Az eget folytonosan nagy területen borítja.



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

Ezen kívül beszélhetünk még függőleges felépítésű felhőkről. Ezek vertikális mérete a troposzféra jelentős részére kiterjed, több rétegen is átível. Rendszerint csapadék hullik belőlük, általában nagymértékben. Ide a Cumulonimbus és a Nimbostratus sorolható.

A felsorolt fajták megállapítása a gyakorlott észlelők számára is rendkívül bonyolult feladat. Ennek főbb okai az OMSZ dokumentumai alapján: [1]

- A felhők megjelenése rendkívül változatos.
- Nem ritka az egyes felhőfajták egymásba alakulása.
- Másképp néz ki egy felhő kék ég alatt, és másképp, ha másik árnyéka vetődik rá.
- Egyes felhők más felhőbe ágyazva, vagy alatta lévő által részben eltakarva jelennek meg.
- Másképp látjuk ugyanazt a felhőt, mint szürkületkor. Éjjel nem minden felhő azonosítható, egyes felhők nem is láthatók.

Viszont észrevehető, hogy bizonyos felhők főbb tulajdonságai megegyeznek. Ilyen módon a listát két csoportra bonthatjuk: Stratus és Cumulus típusú felhők (lásd 2.4. ábra). Az elsőbe tartozik többek között a Stratus, Cirrus, Nimbostratus, Altostratus, a másodikba pedig a Cumulus, Cumulonimbus, Altocumulus. Programunkban jelenleg 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 vizsgálattal. 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, szálas szerkezetű, általában világos, míg

a Cumulus rendkívül változatos sötétségű lehet, nagy önárnyékkal, gomolyos szerkezettel.

Tehát a borultságból és a színezetből már nagy pontossággal megállapítható, hogy a képen látható felhők melyik csoportba sorolhatók. Azonban van még egy probléma. Előfordulhat, hogy 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 PROBLÉMA ELEMZÉSE

Első lépésként a felhőket kell detektálnunk, mivel égtől való elkülönítésük a borultság és a típus megállapításához is szükséges. Ennek megvalósítására többféle megoldási lehetőséget is figyelembe vettünk, majd ezek közül választottuk ki a legmegfelelőbbet. Habár elsőre egyszerűnek tűnhet, jobban megfigyelve bonyolult feladat, mivel mind a felhők és az ég színe rendkívül sokrétűen változhat. Ezt tovább nehezíti az éjszakai megfigyelés lehetősége, illetve a különböző napszakok vizsgálata.

Ha sikerült detektálnunk a felhőket, a kapott értékek alapján már egyszerű számolással kifejezhetjük a nyolcadokban mért arányt, ami a borultságot jelenti. Egyetlen kamerával a kapott érték nem lesz minden esetben teljesen pontos, mivel a kép nem fedi le az egész eget. Ahhoz külön berendezés készítése lenne szükséges. Azonban célunk a már meglévő kamerák felhasználása volt, így ezzel a lehetőséggel dolgozunk. Manapság gyakori megoldás a radarképek használata, ami nagy területre vonatkozóan pontos eredményt ad, azonban drága, és egy bizonyos település egy radarképen már túl kis terület, így az érték pontos megállapítására nincs lehetőség. Ezen kívül a felhők típusának megállapítására sem alkalmas.

A borultságból kiindulva tovább vizsgálhatjuk a felhőzetet, és megállapíthatjuk annak típusát. Azonban a típus pontos megállapításához rendkívül sok tényezőt kellene figyelembe venni, amire egyetlen kamerával nincs lehetőségünk. Így a borultság mellé még a felhőzet színezetét vizsgáljuk, amit mivel már tudjuk a képen a felhők pozícióját, könnyen megkaphatunk. Ezen értékeket a fenti szempontokat figyelembe véve kell összehasonlítani, és megállapítani a legvalószínűbb értéket a felhőzet típusára. Az algoritmus hátránya ilyen módon, hogy egy képen csak egyetlen típust tud megállapítani, míg a valóságban a felhők több rétegben is elhelyezkedhetnek. Ezt jelenleg nem tekintjük problémának, mivel az észlelők feljegyzéseiben is minden esetben csak az "uralkodó" felhőzet típusa szerepel.

2.4 HASONLÓ RENDSZEREK

Habár a teljes rendszerünk összetettségével rendelkező rendszert nem találtunk, felhők detektálásával foglalkozók már az 1990-es évek óta vannak forgalomban [3]. Azonban ezek többsége nem egyszerű kamerát használ, hanem olyan szerkezetet, mely segítségével a teljes égbolt belátható.

Davis 1992-ben publikált egy rendszert, ami szkennelt képeken képes felhőket detektálni. Algoritmusának alapja az azonos színű és szerkezetű részek szegmentálása. Feister 2000-ben jelentette be rendszerét, ami a korábban Shields által fejlesztett algoritmus javított változata. Alapja a kék, vörös és infra sugarak vizsgálata, segítségével éjszakai detektálás is végezhető. Azonban mivel a mi rendszerünk nem alkalmas ezek figyelésére, így ez a megvalósítási forma számunkra nem lehetséges. A legújabb fejlesztés a Yankee Environment System által fejlesztett TSI (total-sky imager). Ez nem csak egy algoritmust, hanem egy teljes rendszert takar, eszközzel együtt. Képes teljes eget lefedő színes képek készítésére, és élő detektálást biztosít. 2003-ban elkezdték a kísérletezést az UV spektrum használatával. [3]

A fent említett rendszerek közül az infra vagy UV sugarakat vizsgálók egyszerű kamerával nem megvalósíthatók, így számunkra nem megfelelők. A szegmentálás lehetne egy lehetőség, azonban Davis rendszerének megvalósításáról nem található dokumentáció. Így a továbbiakban olyan algoritmusokat vizsgáltunk, melyek valamilyen egyszerű kamera képein is végrehajtható küszöbölést alkalmaznak, nem igényelnek speciális felszerelést.

2.5 BORULTSÁG VIZSGÁLATA HIBRID KÜSZÖBÖLÉSES ALGORITMUSSEL

Felhők detektálására az elsőként megvizsgált lehetséges megoldás a Famona és Asano művében [4] ismertetett módszer, mivel több esetben eredményesebbnek bizonyult más intelligens küszöbölő eljárásoknál. A küszöbölésnek két módja lehetséges: lokális- és globális küszöbölés. Előbbi a teljes képre azonos küszöbértéket használ, míg utóbbi meghatározott méretű ablakok alatt vizsgálja végig képünket. A globális módszerek megvalósítása egyszerűbb, így jobban elterjedtek. Ezek továbbfejlesztésére több hisztogram vizsgálaton alapuló technikát is kifejlesztettek. Az általános küszöbölő alkalmazások legnagyobb hibája, hogy nem képesek információt kinyerni a kép felépítéséből, minden képet azonos módon kezelnek. Az alább ismertetett módszer ennek kiküszöbölésére egy hibrid küszöbölést ismertet, amely az információ kinyerésre éldetektálást használ, küszöbölésre pedig az úgynevezett P-tile módszert. Ilyen módon pontosabb értéket kaphatunk az objektumok alakjára, mint egyszerű küszöbölő algoritmusokkal.

A P-tile szó a "percentile" szóra utal, amiből látszik, hogy százalékos értékekkel számol. A módszer lényege, hogy a küszöbértéket úgy állítja be, hogy a felette lévő

pixelek aránya a lehető legközelebb legyen egy előre megadott százalékos értékhez. Ebben az esetben mindig szürkeárnyaltos képpel dolgozunk, így a mis esetünkben mindenképpen előfeldolgozással kell kezdenünk. Az algoritmus feltételezi, hogy az objektum világosabb a háttérnél, és az objektumok a kép egy bizonyos fix százalékát foglalják el. Ezt az értéket jelölhetjük $P\%$ -kal. Mivel tudjuk az objektum arányát a háttérhez képest, pontosabban megtudjuk határozni azt a képen. A küszöbölő algoritmusunk addig változtatja a küszöbértékünket, amíg a lehető legpontosabban el nem érjük a keresett $P\%$ értéket. Ennek elvégzése egyszerű, bármilyen méretű objektumra alkalmazható, és jó a zajszűrő képessége. Azonban látszik az is, hogy nem alkalmazható, ha $P\%$ ismeretlen, vagy képről-képre változik. Mivel ez az információ az esetek többségében nem áll rendelkezésünkre, megoldást kell találnunk annak kinyerésére a képből.

A módszer második része az éldetektálás, amit gyakran használnak a képből való információ kinyerésére, objektumok elkülönítésére. Az élek keretet adnak az objektum(ok) és a háttér között. Módszere, hogy a hirtelen intenzitás változásokat figyeli a képen. Élkeresés eredményeként egy éltérkép (edge map) jön létre. Az élkeresésnek számos algoritmus ismert, de alapvetően két típusba sorolhatók: gradiens alapú módszerek és Laplace megközelítés. Gradiens módszernél lokális maximum értékeket keresünk, így az első deriváltat használjuk, míg Laplace-nál zérus átmeneteket a totális második deriváltban. Néhány ismertebb algoritmus a Roberts, Prewitt, Sobel és Canny éldetektálók. Nekünk a lehető legpontosabb algoritmusra van szükségünk, aminek 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 ne keltsen hamis éleket

Egy számunkra megfelelő algoritmus a Canny éldetektálás [5], mivel ez a fenti kritériumok mindegyikét kielégíti. A feldolgozás öt lépésből áll:

- Elmosza a képet, így csökkenti a zajok mértékét.
- Gradiens értékek meghatározása, kiugró értékek keresése.
- Lokális maximum értékek kiválogatása a maximumok közül.
- Különálló, más élekhez nem kapcsolódó találatok elnyomása, vékonyítás.
- Hiszterézises küszöbölés az élek meghatározására.

Az elmosás Gauss szűrő alkalmazásával zajlik [6]. Ezek után az algoritmus Sobel-operátort használ a gradiens értékek meghatározásához. Ez alapján az élek detektálása már elvégezhető, azonban ezek gyakran szélesebbek, így több él detektálható egyetlen valós helyett. Ennek kezelésére az algoritmus kiszámítja és eltárolja az élek irányát is. Ezek után kiválasztjuk a lokális maximumokat, minden más találatot töröl. Ennek célja, hogy az elmosott élek helyett éles eredményeket kapjunk. A megmaradt értékeket már

nagy valószínűséggel tekinthetnénk valósnak, de akadhat még, amit zaj okozott. Ezek kiszűrésére használhatnánk egy egyszerű küszöbölést, de a Canny algoritmus egy dupla küszöbölést alkalmaz, két küszöbértéket határoz meg. A magasabb felett lévő éleket erősnek, a kettő közöttieket gyengének jelöli, míg az alsó alattit elnyomja. Az erős éleket egyből hozzáadhatjuk az élterképhez, azonban a gyengéket csak akkor, ha csatlakoznak erős élhez.

Persze más algoritmusok is használhatók, de minden esetben célszerű zajsűréssel kezdeni. Az egyik legismertebb és leggyakrabban használt ilyen szűrő a Gauss szűrő [6], amit a Canny algoritmus is tartalmaz.

Ha sikerült a megfelelő éldetektálást elvégezni, akkor megkaphatjuk a $P\%$ értékét. Ennek módja az, hogy az eredeti kép élterképéből kivonjuk a küszöbölt kép élterképét. Ha ezt minden $P\%$ értékre elvégezzük, akkor a kapott értékekből megkaphatjuk a legmegfelelőbbet ott, ahol ez a különbség a legkisebb volt. A két kép kivonását az MSE-vel (Mean Squared Error) végezzük.

A függvényben szerepel egy *Step* érték, aminek változtatásával növelhetjük, illetve csökkenthetjük a pontosságot. Ha nagy értéket adunk neki a programunk gyorsabb, de pontatlanabb lesz, míg kis érték esetén pontosabb, de lassabb. 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 [7].

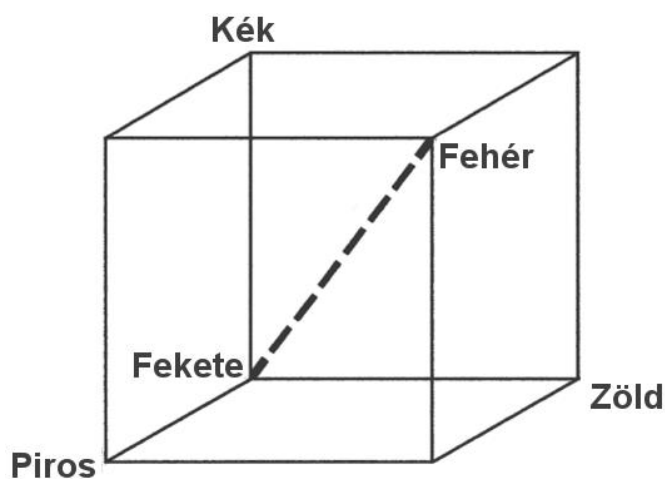
Ezzel a megoldással azonban felhők esetén nem sikerült pontos eredményt elérni. Az első probléma, hogy mindig keres küszöbértéket, míg a mi esetünkben felhőtlen, vagy teljesen felhős ég esetén csak előtér vagy háttér van, tehát nincs szükségünk küszöbölésre. Ezen kívül egyetlen felhő is sokféle színárnyalatot felvehet egyszerre, így ilyen esetekben ez az algoritmus a felhőt is több objektumra bontja (lásd 2.5. ábra). Gyakori probléma, hogy mivel a felhőzet és az ég színe is rendkívül széles skálán mozog, így küszöbölés után a változó színek miatt nem mindig a felhő lesz az előtér az ég pedig a háttér, ami lehetetlenné teszi a borultság pontos számítását egyszerű küszöbölő algoritmusokkal, ugyanis az értékek ebben az esetben felcserélődnek a tudtunk nélkül. Így rájöttünk, hogy hagyományos küszöbölő módszerek segítségével nem tudunk megfelelő eredményt elérni. Új megoldásra van szükségünk.



2.5. ábra - A hibrid küszöbölést és éldetektálást alkalmazó algoritmus hibás működése teljes eget borító, változatos színű felhők esetén. Látható, hogy ilyen esetben is határoz meg küszöbértéket, ami hibás küszöbölést eredményez.

2.6 BORULTSÁG VIZSGÁLATA SZATURÁCIÓ MÉRÉSSSEL

Létezik egy módszer, amely kifejezetten felhők detektálására lett kifejlesztve, ez pedig a szaturáció méréséből való következtetés. A módszer a [3] cikkben került ismertetésre, amely megvalósításával és alkalmazásával nagy előrelépést értünk el. E szerint a felhők dinamikusán 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.



2.6. ábra - RGB kocka a kék, zöld és vörös értékek ábrázolására [3].

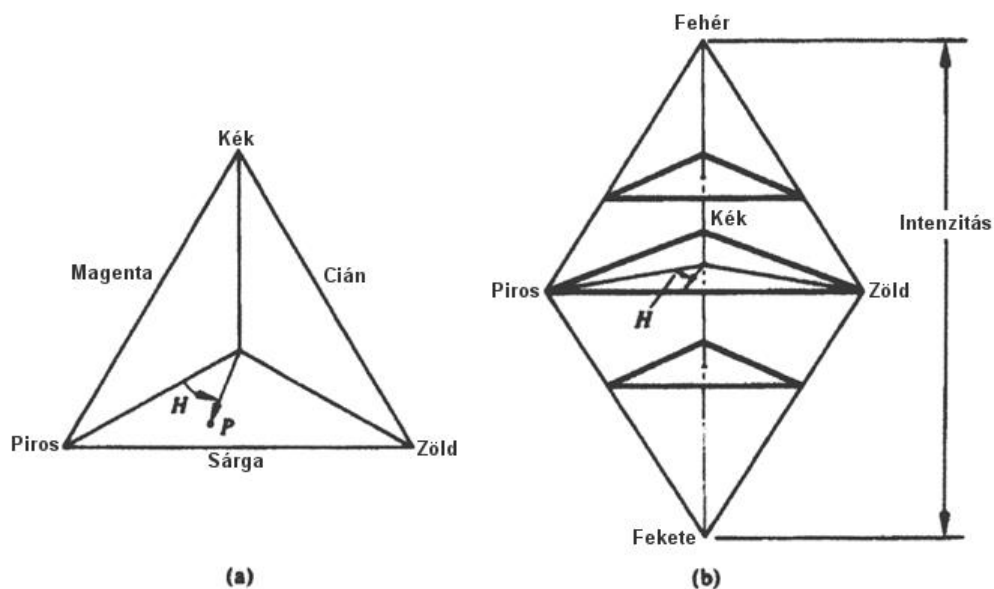
Az RGB modell széles körben elterjedt (lásd 2.6. ábra). Ez a színek és a szaturáció egy konstans fényerőn vett reprezentációja. Azonban mind a felhők és az ég színe rendkívül széles skálán mozog, így nem célszerű ezt a színteret választanunk. Az IHS rendszerben az intenzitás (I) a teljes energiát jelöli az összes hullámhosszon, ami eléri a szemet (lásd 2.7. ábra). Ez felelős a fényerő érzékeléséért, számítási módja az RGB modell alapján:

$$I = \frac{1}{3}(R + G + B) \quad (1)$$

A "Hue" (H) a fény elnyelését, visszaverődését adja meg, így ez felelős a tárgyak színéért, magyarul színárnyalat. Számítási módja:

$$H = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2 + (R-G)(G-B)]^{1/2}} \right\} \quad (2)$$

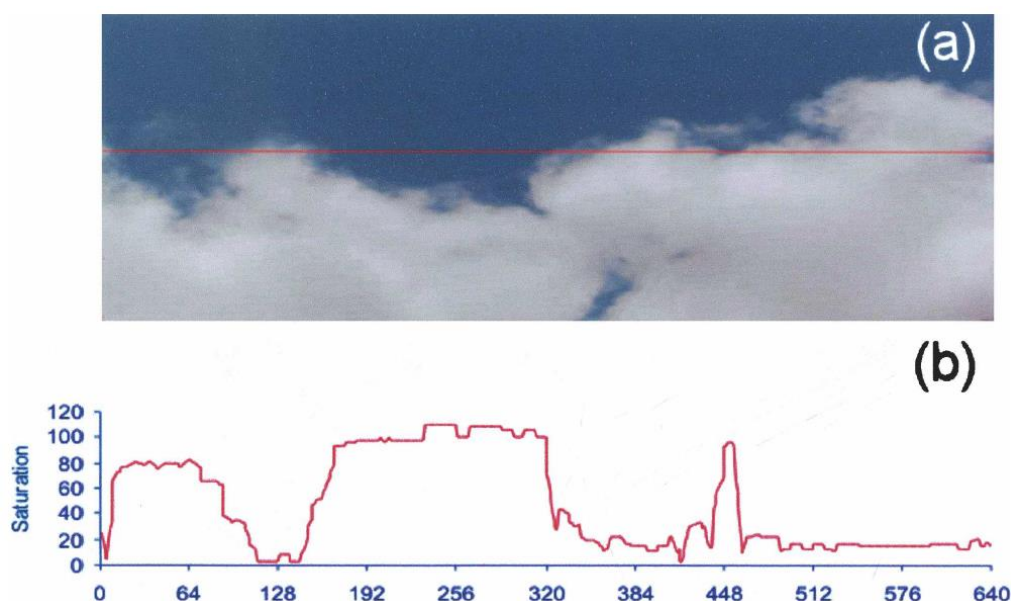
A szaturáció (S) a színtelítettség, ami a színek tisztaságát jelöli. A magas szaturációs értékekre azt mondják, hogy tiszta, mint a derült ég. Az alacsony értékűek olyanok, mint a felhők.



2.7. ábra - Az IHS színtér ábrázolása. (a) Színháromszög, melynek pontjai a zöld, vörös és kék értékek. (b) Árnyalat, telítettség és intenzitás kombinációja háromdimenziós színtérben [3].

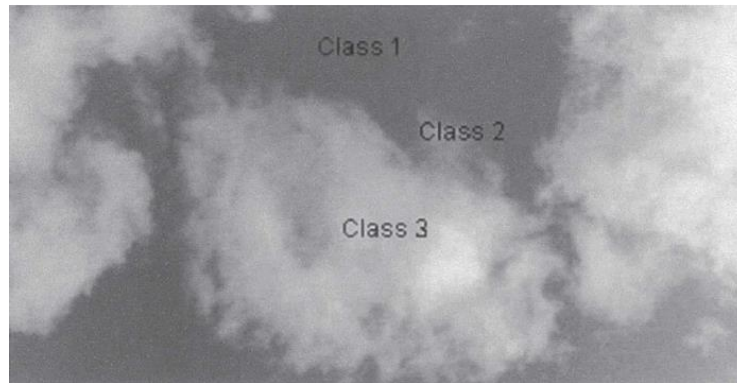
Láthatjuk, hogy a felhőknek jó a fényvisszaverő képessége, általában fehérek, de számos színárnyalatot felvehetnek. Ezzel szemben a kék színű derült ég magas szaturációt eredményez (lásd 2.8. ábra). Ezért az IHS rendszerből mi a szaturációt (S) fogjuk használni algoritmusunk alapjául. Kiszámításának egy lehetséges módja, ha a vörös (R), a zöld (G) és a kék (B) összetevőt ismerjük:

$$S = 1 - \left(\frac{3}{R+G+B} \right) * \min(R, G, B) \quad (3)$$



2.8. ábra – (a) Eredeti kép, amin a piros egyenes jelzi a vizsgált tartományt. (b) Szaturációs értékek az eredeti képen található vonal egyes részein. Jól megfigyelhető az ég és felhő értékei közti különbség [3].

A szakirodalom alapján a képet nem 2 osztályra bontjuk a küszöböléssel, hanem hármat kell meghatároznunk. A pixelek besorolására szaturációs értékük alapján történik (lásd 2.9. 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 szakemberek már megállapították [3], 29 különböző felhőzetet tartalmazó kép feldolgozásával, és azokon a megfelelő értékek meghatározásával (lásd 2. melléklet). Az algoritmust meteorológusok segítségével tesztelték. A valós idejű megfigyelés nem volt számukra megoldható, így képeket adtak a szakembereknek, amiken ezek után 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. Azonban itt meg kell jegyezni, hogy ez nem a tényleges borultságra vonatkozó mérés – a kép nem fedi a teljes eget – csupán a képen látható területre.

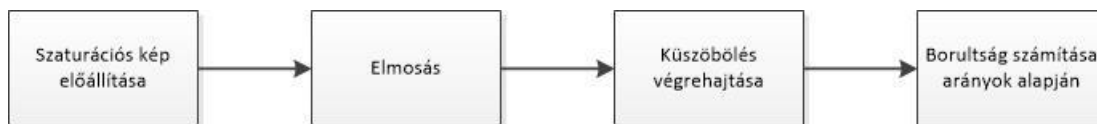


2.9. ábra – Osztályok a pixelek besorolására. Class 1: tiszta ég; Class 2: nem meghatározott; Class 3: felhő [3].

2.7 BORULTSÁG VIZSGÁLAT MEGVALÓSÍTÁSA

A fent leírt két módszer közül a szaturációs mérés bizonyult eredményesebbnek, így ezt a megoldást implementáltuk programunkban. Ebben az esetben kiküszöböltük a hagyományos dinamikus küszöbölések azon hibáját, hogy mindenképpen keresnek küszöbértéket. Ezen kívül a több osztályba való besorolást is eredményesnek találtuk, ugyanis így még több információ nyerhető ki a képből, ezzel is javítva a feldolgozás eredményét.

Az algoritmusban (lásd 2.10. ábra) első lépésként a szaturációs érték megállapítását kell elvégeznünk minden pixelre, ami alapján egy úgynevezett szaturációs térképet kapunk. Ennek kiszámítására több lehetséges képletet is kipróbáltunk, de a fent említett adta a legbiztosabb eredményt. Más módszereknél gyakran erősen ki lettek emelve az élek, amire nekünk jelenleg nincs szükségünk, rontja a küszöbölés pontosságát. Az alkalmazott képlet segítségével megfelelő eredményt kaptunk, azonban számolnunk kell azzal az esettel, amikor a pixel vörös, zöld és kék értékei egyaránt nullák. Ennek kiküszöbölésére ilyen esetekben a vörös értéket eggyel helyettesítjük a képletben, ami még nem eredményez észrevehető eltérést a végeredményben. Azonban a szakirodalom [3] által meghatározott értékek alkalmazásával a különböző szintek határait nem kaptunk pontos eredményt. A probléma megoldására a tesztek során folyamatosan mértük a küszöbértékeket, és ez alapján próbáltuk meg a lehető legpontosabb értékeket meghatározni. Azonban 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, amelyek alkalmazásával gyorsan és pontosan állíthatjuk be a küszöbértékeket.



2.10. ábra – Borultság vizsgálat folyamata

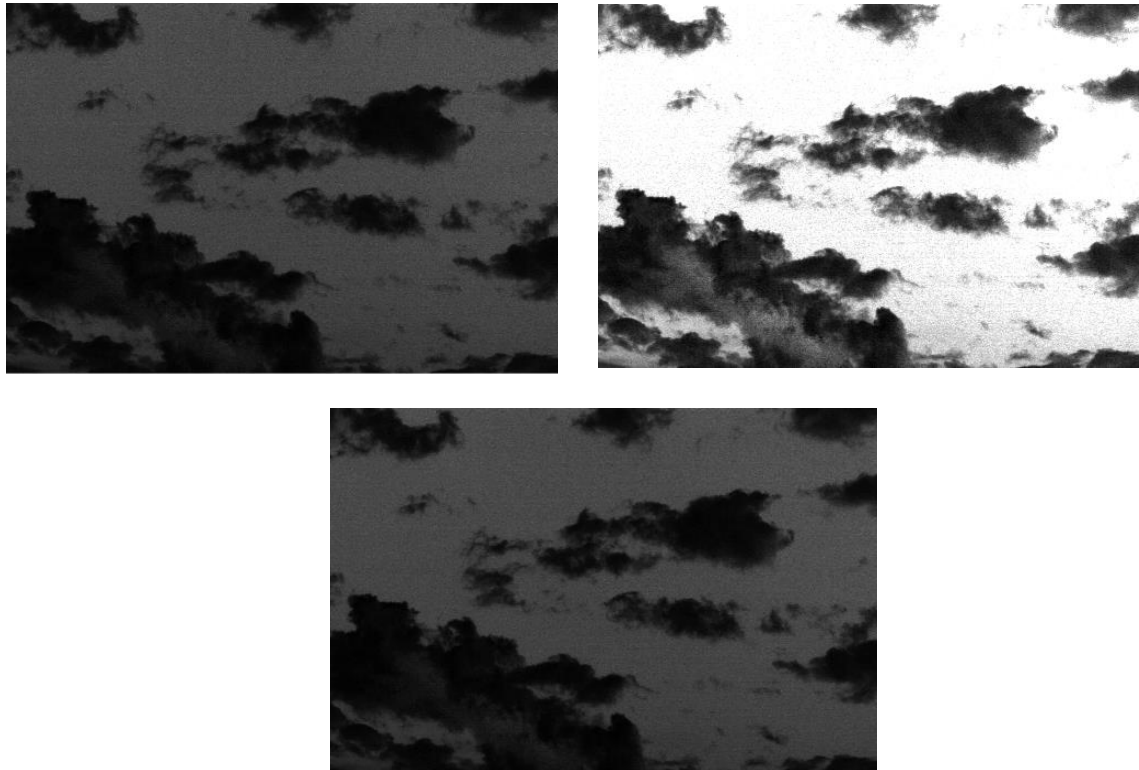
Automatikusan a program előállítja a szaturációs képet (célszerű olyan képen végezni a konfigurálást, amin a felhő és ég jól elkülöníthető), ezen egy Otsu binarizálást végez, ami megad egy küszöbértéket, majd ez alapján az érték alapján két határt ad meg. A korábban már említett okok miatt ez a megoldás a borultság vizsgálatára nem alkalmas, azonban egyetlen általunk megadott képen pontosan meg tudjuk határozni a küszöbértékeket a segítségével. A binarizálás végeredményéhez képest a két határt egy-egy, a tesztek alapján megkapott fix érték hozzáadásával, illetve kivonásával kapjuk. Azonban a szakirodalom által meghatározott módszerrel, miszerint a teljes tartományt 5 részre bontjuk, amiből mi hármat használunk, nem sikerült pontos eredményt elérni. A különböző zajok miatt gyakran kaptunk olyan értéket, ami az általunk kijelölt 3 osztályból egyikbe sem tartozott. Ennek kiküszöbölésére az ég felső- és a felhők alsó korlátját kihagytuk. Így végeredményképpen az automatizált megoldással gyors és pontos határértékeket kaptunk.

Ha az automatikus konfigurálás értékét mégis pontatlannak találjuk, lehetőségünk van manuális beállításra. Ebben az esetben, egy új ablak nyílik meg, melyben maximum 3 képet nyithatunk meg. Ezekre célszerű egy Cumulus-, egy Stratus- és egy vegyes típusú felhőzetet tartalmazó képet beállítani. A megnyitott képeken 2 csúszka és valós időben végzett küszöbölés segítségével állíthatjuk be a határokat. A betöltött képek méretét csökkentettük. Ennek eredményeként a feldolgozás pontatlanabb lett, de nem annyira, hogy a konfigurálás pontatlan legyen, viszont nagymértékben felgyorsult, így a beállítást gördülékenyen végezhetjük. Így mindig jól látható a változás, és egész pontosan be tudjuk állítani a küszöbértékeket a nekünk megfelelőre. A konfigurálás a rendszer működése közben is, bármikor elvégezhető.

Azonban az előzetes tesztek alatt azt tapasztaltuk, hogy az előállt szaturációs térkép még nagymértékű zajt tartalmaz, így valamilyen előfeldolgozásra is szükségünk lesz folytatás előtt. Így elengedhetetlen lépés még az előállt szaturációs képen végzett elmosás. A megfelelő algoritmus segítségével a képet sokkal tisztábbá tehetjük, ezzel nagymértékben fokozva a pontosságot. Több módszert is kipróbáltunk, köztük a Gauss módszert, de messze a legjobb eredményt az AForge "Blur" algoritmusával kaptuk (lásd 2.11. ábra).

Konfigurálás után már megfelelő pontossággal detektálhatjuk a felhőket. Ezt úgy végezzük, hogy a beállítások után kapott két küszöbérték, valamint a 0 és 255 értékek határokként való alkalmazásával a kép minden pixelét egy-egy csoportba soroljuk. Az eredeti képről egy másolatot készítünk, majd a vizsgált képpont színét, a csoport értékének megfelelően módosítjuk, így a későbbiekben egyértelműen azonosítani

tudjuk. A három színnek a kék, zöld és fehér értékeket választottuk, melyek ebben a sorrendben az eget, nem meghatározottat és a felhőket jelölik (lásd 2.12. ábra). Ezek után következik a küszöbölés utáni 3 szint arányainak összehasonlítása. A kép méretéből meghatározható az összes pixel mennyisége, amit nyolccal elosztva megkapjuk az egy okta mértékét. A felhőnek detektált pixelek számát ezzel az értékkel elosztva megkapjuk, hogy az hány oktának felel meg. Ennek számításánál végeztünk teszteket felfelé illetve lefelé kerekítéssel is, de a legpontosabb eredményt úgy kaptuk, ha az érték kerekítését mindig a matematikai szabályoknak megfelelően végeztük.



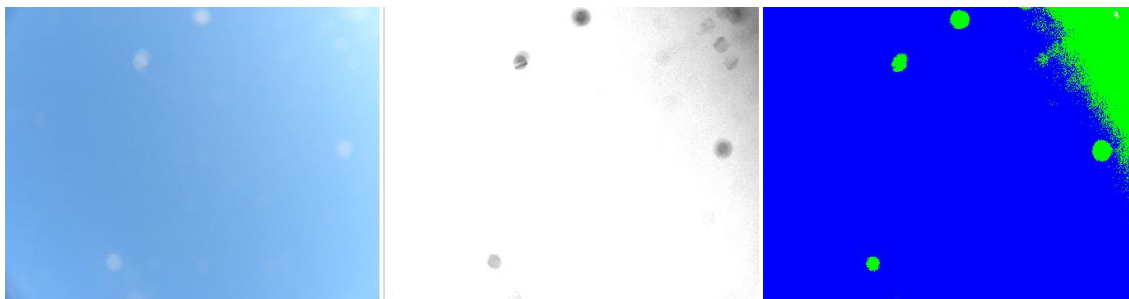
2.11. ábra – Bal képen az elmosás nélküli szaturációs térkép látható, míg a jobb az AForge Blur algoritmusát, az alsó pedig a Gauss módszert alkalmazza.



2.12. ábra – Eredeti kép, szaturációs kép, küszöbölt kép – kék: ég, fehér: felhő, zöld: nem besorolt

Hogy a feldolgozás pontosságát tovább növeljük, célszerű kihasználnunk, hogy nem kettő, hanem három osztály áll a rendelkezésünkre, és figyelembe kell vennünk, hogy előfordulhatnak speciális esetek is. Ilyen a felhőtlen ég, illetve a már korábban külön kiemelt 1 és 7 oktás esetek, melyeknél a lyukak vagy felhők mennyiségét kell figyelnünk. Megoldásként az arányok számolását nem egyszer, hanem kétszer végezzük el. Először csak a biztosan felhőnek detektált részeket számítjuk "Felhő" kategóriába, míg második számítás esetén a "Nem meghatározott" részeket is. Ha az elején 0 oktát kapunk eredményül megvizsgáljuk, hogy a kép századánál több vagy kevesebb felhőt találtunk-e. Ezt az értéket számos teszt után választottuk, hogy a rosszul detektált zajok kiszűrésével pontosabb eredményt kaphassunk. Ha kevesebb a mennyiség, akkor 0 oktát állapíthatunk meg, és befejezzük a számítást, ha nagyobb, akkor folytatjuk a vizsgálatot. Megnézzük, hogy az érték 8 okta volt-e. Ha igen, akkor azt hasonló módon, de az ég mennyiségét figyelve vizsgáljuk meg a képet. Ha a meghatározott aránynál kevesebb mennyiségű lyukat találtunk, akkor megállapíthatjuk, hogy 8 okta a borultság. Ha nem, akkor a feldolgozást tovább folytatjuk, a második módszerrel is kiszámoljuk az értéket. Ebben az esetben már nem végzünk további vizsgálatokat. Az algoritmus helyes működése esetén abban az esetben sem merül fel probléma, ha teljesen felhőtlen, a Nap által megvilágított ég szerepel a képen (lásd 2.13. ábra), ugyanis ezt "Nem meghatározott"-nak detektáljuk, amit első alkalomnál nem veszünk figyelembe, ha nem található felhő az égen.

A kapott értékekkel jó eredményt kaptunk a teszteken, azonban ezeket a későbbiekben még a felhők típusának megfelelően változtathatjuk.



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

2.8 FELHŐK OSZTÁLYOZÁSA KÉPFELDOLGOZÁSSAL

Felhő típusának megállapítására leggyakrabban radarképek [2] alapján tesznek becsléseket, azonban ez költséges és egyetlen konkrét, kis területre nem alkalmazható. Az algoritmust így az észlelőktől, illetve szakirodalmakból szerzett információk alapján magunknak kellett kialakítanunk úgy, hogy egyszerű kamerákkal is működőképes legyen. A felhők sikeres detektálásából és a mért borultságból kiindulva próbáltuk megállapítani a felhők típusát is (lásd 2.14. ábra). A módszernek szüksége van paraméterként az eredeti képre, a küszöbölt képre és a már korábban kiszámított borultság értékre. Első lépésként az eredeti képet vizsgálva egy hisztogramot kell készítenünk, de csak azokat a részeket vizsgálva, ahol felhő található. Ehhez maszkként használhatjuk fel a küszöbölt képet. A hisztogram készítéséhez minden pixel esetében a három érték (vörös, kék, zöld) átlagát vesszük figyelembe, ezt tekinthetjük szürkeárnyalatos képnek is. Ezek után össze kell számolnunk, hogy hány intenzitás értéket találtunk összesen. Ezeken az értékeken célszerű egy szűrést alkalmazni, hogy a zajokat kiszűrjük. Ezt úgy tettük meg, hogy csak a bizonyos mennyiségű pixelt tartalmazó hisztogram értékeket vettük figyelembe. Innentől kezdve minden szükséges információ a rendelkezésünkre áll a típus meghatározásához. Tudjuk a felhő színeinek mennyiségét, valamint a borultságot. Több kép vizsgálatával megállapítottunk egy küszöbértéket a színek mennyiségére, ami felett a felhő Cumulus, alatta pedig Stratus. Ez lesz az elsődleges szempontunk a vizsgálatnál. Azonban a 7 okta alatti, és az e feletti értékekkel külön számolunk. Ha alatta van a borultság, tovább vizsgáljuk az értékeit. Ha 0, akkor tiszta az ég, nem található felhő, így az intenzitások számával nem kell foglalkoznunk. Ettől eltérő esetben a színek számának megfelelően döntünk. Ha egy előre definiált határérték fölött van, akkor Cumulus típusú a felhő, míg ha alatta, ismét a borultságot vizsgáljuk. Ha 1 okta az érték, akkor Cumulus, míg ettől eltérő esetben Stratus a felhő típusa. Ha a borultság 7 vagy 8 okta volt, akkor elegendő csak a színek számát vizsgálnunk. Ha a már fent is említett határ feletti az érték Cumulus, míg az alatti esetén Stratus típusú a felhőnk.

Ha ez alapján megállapítottuk a típust, meg kell vizsgálnunk azt is, hogy a kapott érték megfelel-e a korábban kiszámított borultságnak. Tudjuk, hogy 8 okta borultsághoz nem tartozhat Cumulus típus, így amennyiben nem felelnek meg egymásnak az értékek, az elvárásnak megfelelően módosítanunk kell [1] [2]. A felhő típusánál nem jönnek elő az egyszerű, kisméretű fényképből adódó hibák, míg a borultságot befolyásolhatja, így ezt az értéket korrigáljuk, és a típust vesszük helyesnek. Ezt az észlelők a gyakorlatban is így végzik. Előfordul, hogy az eget teljesen elfedi egy Cumulus típusú felhő, de ekkor feltételezik, hogy lyukak találhatók, csak nem láthatók, és az érték így 7 okta. Mi ezt az esetet a borultság megállapításánál már lekezeljük, így itt ezzel nem kell foglalkoznunk.

2.9 SPECIÁLIS ESETEK KEZELÉSE

Az elkészített algoritmussal számos tesztet végeztünk. A kapott hibás 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 szakemberek által megállapított értékek többségében megegyeznek a program számított értékeivel, de a teljes ég vizsgálatával készített hivatalos dokumentumok adatai a képek alapján megállapítottaktól eltérhetnek. Ennek javítására készíthetnénk egy rendszert, ami az egész eget figyeli, ehhez hasonló gépekkel a hasonló rendszerek elemzésekor is találkoztunk, 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 sorozatfelvételekkel hasonló hatást érünk el, mintha a kamerát mozgatnánk, és 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 értékeit, 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, mivel több képen kevésbé valószínű, hogy ugyanolyan módon előfordulnak ugyanazok a hibák.



2.14. ábra – Felhő típus detektálásának folyamata

A kamerák a nap 24 órájában készítenek képeket, de az éjszakai képeken felhők már nem láthatók a sötét miatt. Időponthoz a sötétedés nehezen köthető, mivel az folyamatosan változik. Használhatnánk internetről kapott információt a naplemente és napfelkelte időpontjairól, azonban helységtől és pozíciótól függően még ez is relatív, így célszerű ennek vizsgálatára is képfeldolgozást alkalmaznunk. Programunkban ezt az esetet is le kell kezelnünk. Erre megoldásnak egy algoritmust készítettünk, mely a kép szürkeárnyaltos intenzitás értékeinek átlagát vizsgálja. Számos képet megvizsgálva azt figyeltük meg, hogy 8 október, teljesen sötét felhőzet esetén is jelentősen magasabb az intenzitások átlaga, mint esténként. Rendelkezésünkre álltak sorozatfelvételek, melyeken sötétből indulunk, és közben kivilágosodik, ezért számos időpontban és

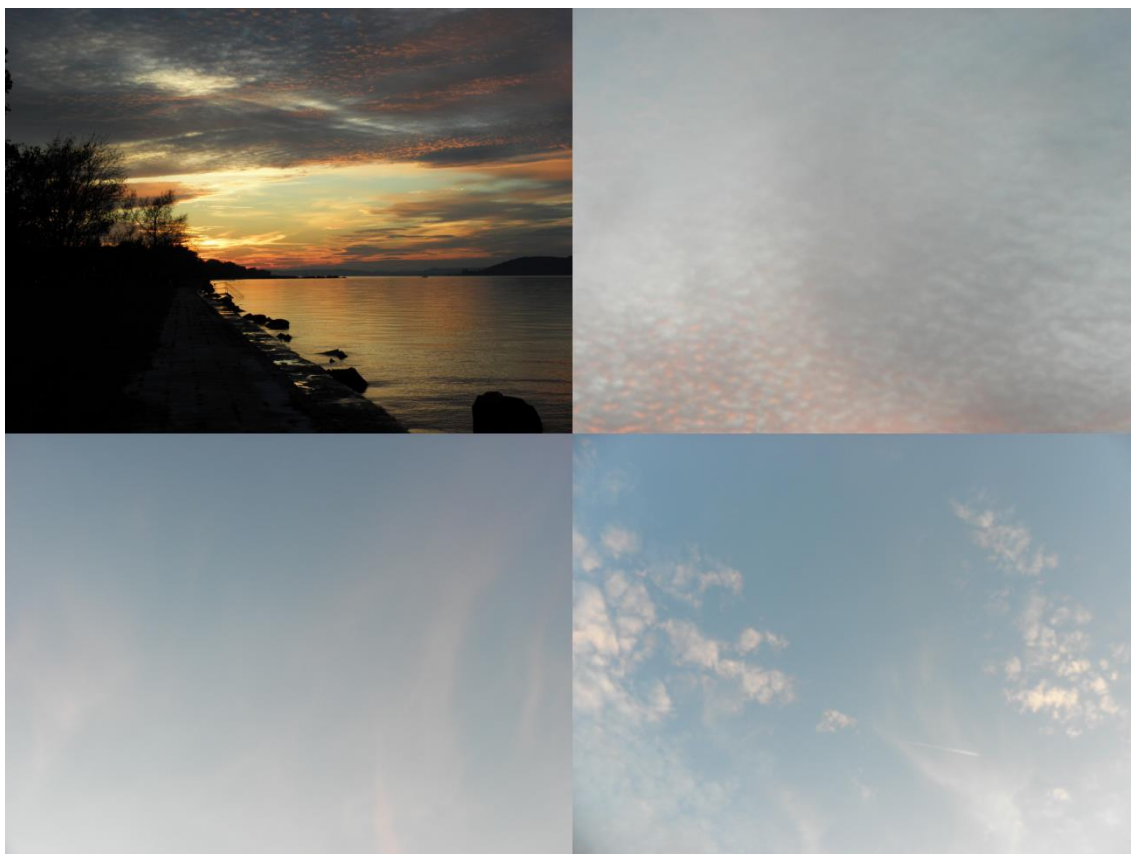
fényviszonyban tesztelhattuk az algoritmust. Így megállapítottunk egy értéket, amin még sikerült a felhőket detektálni, de ami alatt már nem lehetséges. Ezt a vizsgálatot a program minden detektálás előtt lefuttatja, és amennyiben az intenzitások átlaga ez alatt az érték alatt van, akkor jelzi, hogy a detektálás nem lehetséges, és leállítja a folyamatot.

A nappallal és éjszakákkal már egyaránt számoltunk, de figyelembe kell vennünk a köztes állapotokat is. Naplementekor illetve napfelkeltekor a felhők színezete gyakran narancssárgásra változik. Így a felhők szaturációs értéke változhat, így ezekkel az esetekkel is külön teszteltünk. Azonban azt tapasztaltuk, hogy függőlegesen felfelé néző kamera esetén rendkívül ritkán található elszíneződés, és az is olyan kicsi mértékű, hogy a végeredményt nem befolyásolja (lásd 2.16. ábra).



2.15. ábra – A rendszer hasonló, esőcseppekkel borított képek esetén is sikeresen működött. Ez esetben az eredmény 8 oktás Stratus felhőzet.

Felmerült ezen kívül a kérdés, hogy mi történik akkor, ha esik az eső. A kamerát egy bura 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 (lásd 2.15. ábra).



2.16. ábra – Látható, hogy míg horizontálisan elhelyezett kamera képén a naplemente jelentősen elszínezi a felhőzetet, azonos időpontban elkészített képen, felfelé néző kamera esetén ez a jelenség csak minimálismértékben tapasztalható.

2.10 EREDMÉNYEK ÉRTÉKELÉSE

Felhős ég esetén az algoritmus rendkívül pontosan képes az eget elkülöníteni a felhőktől, legyen szó akár sötétebb, akár világosabb féléről. 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, azaz 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.

Az OMSZ képei közül a hivatalos észlelések dokumentumai alapján megközelítőleg 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 véletlenszerű időpontban rögzített képek is készültek. A programban megvalósítottunk egy funkciót, mely több képen egymás után képes elvégezni a detektálást, és a kapott értéket összehasonlítani az észlelők által mértekkel, valamint 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. Az adatokat a 3. és 4. mellékletek tartalmazzák.

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

1. 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

2. 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

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. 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). Korábban említettük az esős képek kérdését. A tesztek alapján az ilyen képek 85.71%-a helyesen lett detektálva.

	Felhő típus esetében
Érzékenység	0,9825
Specifikusság	0,9888
Precizitás	0,9974
Felidézés	0,9825

3. táblázat - Fals negatív illetve fals pozitív értékekből kifejezett paraméterek a tesztek értékelésére.

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, majd ezek alapján megállapítottunk szokásos paramétereket. Az eredményeket az 1–3. táblázatok mutatják.

3 MAGASSÁGI SZÉLIRÁNY MEGHATÁROZÁSA

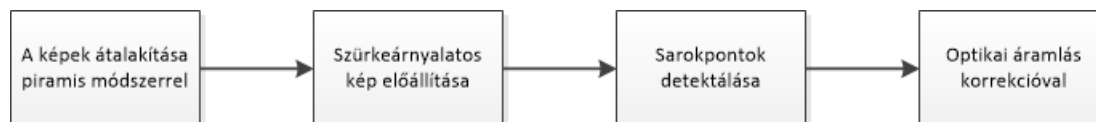
3.1 CÉL MEGHATÁROZÁSA

Az OMSZ szeretné, ha felhőtípusok meghatározása után a felhők mozgásának az irányát is meg tudnánk 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 szél iránya a felhők mozgásából meghatározható. Fontos megjegyezni, hogy ez az érték nem egyezik a lent, gépek vagy észlelők által mért értékkel, mivel a szél erőssége és iránya a föld felszínétől távolodva folyamatosan változik.

Mivel egy képen az égtájak relatívak, valamilyen módon be kell tájolnunk azokat. Erre két megoldás merült fel. Az első, hogy a kamera mindig olyan pozícióban kell, hogy álljon, hogy a kép teteje mindig észak felé nézzen. A második, jelentősen rugalmasabb megoldás, hogy a programban lehetőséget adunk az égtáj konfigurálására. Erre egy mód, hogy a felhasználó megadhatja, hogy a kép teteje milyen irányba néz, és a későbbiekben mindig ehhez viszonyítva adjuk meg a mért értékeinket. A programban a második verzió is megvalósításra kerül, azonban a folytatásban mindig fix értékekkel számolunk a könnyebb áttekinthetőség érdekében.

Figyelembe kell vennünk, hogy a magasság változásával változhat a szél iránya, valamint a felhők is több rétegben helyezkedhetnek el. Ebből adódóan előfordulhat, hogy a képen látható felhők különböző irányba mozognak, mivel más-más szinteken vannak. Első sorban mi egy konkrét irányt szeretnénk meghatározni, több réteg figyelésére nem térünk ki.

Az OMSZ több sorozatfelvételt is készített számunkra (kb. ezer képet), így ezeken a felvételeken kell megfelelő eredménnyel detektálnunk az elmozdulásokat. A sorozatokban Cumulus és Stratus típusú felhők is találhatóak, magasságuk 800–1000m, és 1500–1800m között van. A magassági szél mind a két szinten kb. 15–17 m/s, ám a sebesség meghatározása nem elvárás, ehhez ugyanis a felhők magasságát is meg kéne tudnunk becsülni kép alapján, de ilyen információt nem tudunk kinyerni. Így feldolgozás során csak a képsíkkal párhuzamos összetevőt tudjuk meghatározni (látszólagos szélirányt).



3.1. ábra – A szélirány meghatározásának a lépései.

3.2 PROBLÉMA ELEMZÉSE

A felhők folyamatosan változnak, rendkívül változatos az alakjuk, és színük is. Találhatunk szinte teljesen homogén felhőket (ilyenek például a Stratus típusú felhők), de rendkívül sok intenzitást tartalmazókat is. Méretük is az egészen kicsitől, a teljes képet betöltőig terjedhet. Ebből egyből látszik, hogy nem lesz elég a felhők elmozdulásának meghatározása, speciális eseteket is kezelnünk kell. Ilyen lesz a tiszta ég, illetve a teljesen homogén kép, amin nem látszik az elmozdulás. Ehhez célszerű felhasználni az előző fejezetekben ismertetett algoritmusok eredményeit, ugyanis ez pontosan a borultságot jelenti. 0 okta (felhőtlen ég) esetén nem jelenthetjük ki, hogy nincs magassági szél, azt kell mondanunk, hogy nem tudjuk detektálni. Azonban, ha találtunk felhőt, de az nem mozdult el, akkor az időt biztosan szélcsendesnek jelezhetjük. 8 oktas borultság esetén (teljesen fedett ég) szintén nem jelenthetünk ki egyből semmit, ugyanis ebből még nem következik, hogy nincs mozgás, további vizsgálatra lesz szükség. Ha találtunk felhőt, akkor megadhatjuk az értéket, de ha nem, akkor ismét csak azt mondhatjuk, hogy nem lehet detektálni.

Amint azt már korábban ismertettük, a kamerák 10 percenként fognak képet készíteni, így az algoritmusnak is ennyi ideje lesz a futásra. Ebből következik, hogy az elsődleges szempont nem a sebesség, hanem a pontosság lesz. Minden alkalommal 3–4 kép fog a rendelkezésünkre állni, körülbelül 5–10 másodperc eltéréssel készítve. Ezeken kell meghatároznunk a szél irányát. Ennek megvalósítására a képek egymáshoz való megfeleltetése szükséges.

Eleinte úgy gondoltuk, a felhő alakjának változása problémát fog jelenteni, mivel a képeket nehezebb lesz összepárosítani. Azonban a kapott sorozatképeken végzett tesztek alapján kiderült, hogy a képek készítése között eltelt pár másodperc alatt nem változik nagymértékben a felhők alakja, így ettől a problémától nem kell tartanunk. A megvalósítást így abban láttuk, hogy minden képen jellemző pontokat keresünk, majd a szomszédos képeken megkeressük a detektált pontok párját. Egy-egy pont a képen való pozícióját, valamint a párjához képest mért eltolást figyelve megkaphatjuk a felhő mozgásának irányát, amiből a szél is meghatározható. Azonban vannak tévesen detektált pontok, amik esetleg nem is a felhőhöz tartoznak, és figyelembe kell vennünk, hogy lehetnek tévesen párosítottak is. Így a halmazunkból olyan módon kell meghatároznunk egyetlen végeredményt, hogy a kiugró értékek ne befolyásolják a többséget.

3.3 MEGVALÓSÍTÁS

Abban az esetben, ha a fent leírt módszert sikerül megvalósítani, vagyis a helytelenül detektált vagy párosított részeket ki tudjuk szűrni, akkor nincs szükségünk elő feldolgozásra. Azonban először egy olyan módszert valósítottunk meg, ami megpróbálja a különböző zajok kiszűrését. Ennek érdekében első lépésként egy

előfeldolgozást kell végeznünk (lásd 3.1. ábra). A hibás pixelek kiszűrésére jó megoldás lehet az átlagoló szűrő, ami az aktuális pixel köré helyezett ablakban lévő értékek átlagát helyettesíti a kiválasztott helyére. Viszont, ha a zaj nagyon erős volt, ez a módszer nem garantálja annak teljes eltüntetését. Így teszteltünk medián szűrővel is. A medián számításának módja a statisztikából már ismert lehet, és ez a módszer nincs terhelve a zaj mértékével, azonban elmosta az éleinket. Egy újabb lehetőség a Gauss szűrő alkalmazása, ahol számításnál az aktuális pixeltől való távolságot is figyelembe vesszük [6]. Az eddig kiemelt módszerek közül a legjobb eredményt az utóbbi szűrővel kaptuk, azonban még ez sem vezetett korrekt eredményhez. A 3×3 -as maszkkal még sok hibás pixel maradt a képen. Az ablak méretének növelése bár csökkentette a zaj mennyiségét, de már számunkra értékes információt is veszítettünk.

Egy másik megközelítés a probléma megoldására a képpiramisok voltak ([8] és [9]). Ebben az esetben az eredeti kép különböző skálázású másolataiból épül fel egy-egy kép, ami szintenként kettő hatványai szerint csökken vagy nő. Ahogy a piramisban lefelé haladunk a képünk mérete csökken - gyorsítva ezzel a feldolgozást - és ezzel egyidejűleg simább is lesz, viszont minimális a számunkra lényeges információ vesztesége. Végeredményképpen homogén felületen se tapasztaltunk véletlen sarokpont detektálásokat (lásd 3.2. ábra).



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

Miután sikeresen eltüntettük a zajokat, következő lépés a jellemző pontok keresése. Ezek lényege, hogy maga a pont és annak helye egyértelműen legyen meghatározható. Jellemző pontnak több dolgot tekinthetünk egy képen, például vonalvégződéseket, lokális minimum vagy maximum értékeket, de képfeldolgozásnál gyakori megoldás az is, hogy sarokpontokat keresünk. A megvalósításhoz mi is ezt választottuk. Saroknak két él találkozása tekinthető. Ez azt jelenti, hogy a képfüggvény nagymértékben, több irányban változik. Ha csak egy élünk van, a változás egy irányú. Ennek hátránya, hogy nem megfelelő beállítások mellett rengeteg sarokpontot találhatunk egy képen, viszont előnye, hogy ugyan azt a pontot több képen is azonosíthatjuk akár különböző fényviszonyok, eltolás, elforgatás mellett. Erre egy jó algoritmus a Harris sarokpont detektáló, ami Movarec algoritmusának egy továbbfejlesztett változata (lásd 3.3. ábra).

Ennek segítségével minden képünkön detektáljuk a sarokpontokat. A módszer a pontok köré egy ablakot illeszt, majd a képfüggvény változásának mértékét figyeli, az ablak egy bizonyos irányba való elmozdulásakor.



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

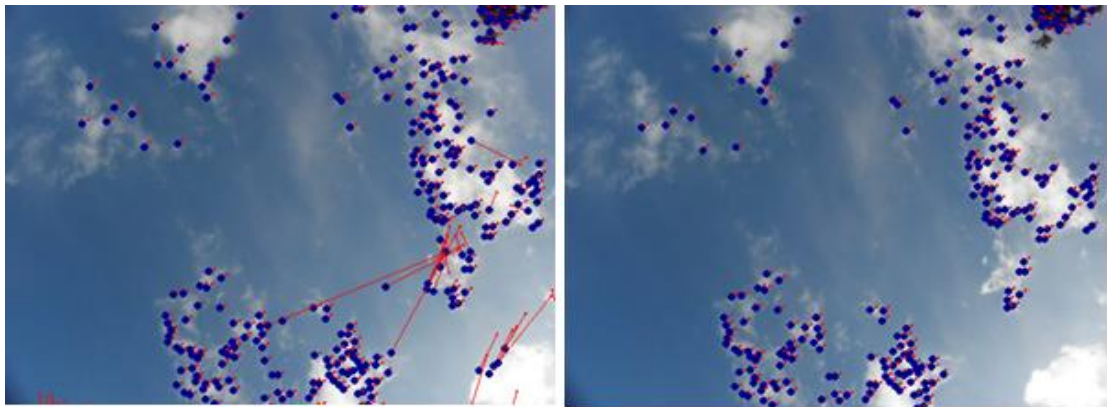
A sarokpontok detektálása azonban önmagában még nem elegendő, meg is kell őket feleltetnünk egymásnak. Erre optikai áramlást használtunk, ami nem más, mint több képen azonos képrészletek megfeleltetése és ezáltal az elmozdulás detektálása. A detektált elmozdulás nem feltétlenül egyezik meg a valós elmozdulással, mivel a valós háromdimenziós világ kétdimenziósra leképezett képein dolgozunk. 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 [10]. Azonban a korábban már ismertetett kamera rendszerben mi tudjuk, hogy a készülék mozdulatlan, így minden a képen történő elmozdulást a felhők mozgásának tekinthetünk. A módszert a piramistechnikával bővített Lucas-Kanade-féle jellemzőkövetéssel valósítottuk meg a [8] és [9] művek alapján. A végeredményen azonban még mindig látható volt, hogy utófeldolgozást igényel. A téves párosításokat el kell dobunk. Tekinthejtük ilyennek a kép méretén túlmutató-, valamint a többitől irreálisan eltérő irányú vektorokat is.

Ezzel a megvalósítással már sok esetben sikerült a helyes detektálás, azonban bizonyos esetekkel nem számol. Nem tudja kezelni sem a túl sok detektált sarokpontot, sem a homogén felületeket. Ezen kívül az algoritmus bonyolult, és lehetne fejleszteni, ha az elmozdulás irányokból zajoktól függetlenül meg tudnánk állapítani a többségben lévő vektorok irányát. Így viszont az előfeldolgozásra nincs szükségünk. A Harris sarokpont detektálás jól bevált, paraméterekkel kellőképpen az adott feladatra hangolhatjuk.

Következő lépés a korreláció vizsgálata lett, amihez az Accord.Net "CorrelationMatching" osztályát használtuk. Ebben az esetben mindig két képet vizsgálunk. Az algoritmus az előzőekben detektált sarokpontokra egy-egy ablakot illeszt, és ebben vizsgálja meg az értékeket. Ha a két képen valamelyik pontokra

használt értékeket kapott, azokat összepárosítja. Ennek végeredményeképpen a képeket megfeleltettük egymásnak. Azonban most kell lekezelnünk a zajok, és a hibás párosítások problémáját (lásd 3.4. ábra).

Megoldás a homográfia használata. Ez egy mátrixot jelent, aminek segítségével információt kaphatunk két kép egymáshoz viszonyított transzformációjáról. Ilyen az elforgatás, eltolás, skálázás. Képfeldolgozás terén gyakran használt módszer, számos területen alkalmazható. Mi a RANSAC (Random Sample Consensus) Homography Estimator-t használtuk a programunkban [11]. Ez az értékek közül a legrobosztusabbat választja ki, így a zajok és hibás értékek a végeredmény befolyásolása nélkül ki lettek szűrve.



3.4. ábra – Az optikai áramlás eredménye korrekciók nélkül (balra), majd megfelelő korrigálások elvégzése után (jobbra) – pirossal az elmozdulások.

3.4 EREDMÉNYEK ÉRTÉKELÉSE

A tesztek az OMSZ által biztosított sorozatfelvételeken végeztük, ami körülbelül 1000 képet jelent. Ezek között található volt Cumulus és Stratus típusú felhő – lehetővé téve a változatos színű és a homogén képeken való tesztelést – valamint esőcseppes képek is.

Az elvégzett teszteken az algoritmus változatos színű felhőkkel jól, míg Stratus típusúakkal rosszabbul szerepelt. Előbbinél az esetek többségében helyesen sikerült a szélirány becslése, a legnagyobb hiba fél égtáj eltérés volt. Homogén felületen azonban az egyértelmű képpontok hiánya miatt nem lehet pontos becslést tenni.

A fent említett eredmények három képből álló sorozatokon végzett tesztek eredményei. Ezt javíthatjuk az egy ciklusban vizsgált képek számának növelésével, így a tesztek elvégeztük tíz képes sorozatokon is. Így Cumulus típusú felhők esetén sikerült elkerülnünk az eddigi minimális eltéréseket is. Az eredmény homogén felületen is javult, de messze elmaradt a változatos felhőzet mellett.

Az algoritmus automatikusan korrigálja saját beállításait, amennyiben a kép túl kevés vagy túl sok detektálható sarokpontot tartalmaz az aktuális konfigurációval. Ez

hasznos funkció, azonban nagy képeken időigényes, és nem minden esetben vezet megfelelő eredményhez (például homogén képek), így figyelni kell, hogy ne alkalmazzuk egy adott határnál többször. Ha ezt átléptük, és túl sok a képpont, a kép méretének csökkentése szükséges, míg túl kevés esetén kijelenthetjük, hogy a kép homogén.

A megfelelő eredmények érdekében célszerű felhasználni a borultság vizsgálat eredményeit, és amennyiben Stratus típusú felhőzetünk volt, "Nem meghatározható"-nak tekinteni az irányt. Ugyan ezt tehetjük esőzés esetén is, mivel a búrán lévő esőcseppek miatt a helyes detektálás lehetetlen, ráadásul csapadékhoz a legtöbb esetben 8 oktás homogén felhőzet tartozik. Ebből adódóan egy újabb témakör az esőzés detektálása, melynek eredményével a fent leírt esetet elkerülhetjük.

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 helyettesíthetők 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 adóttak. Az OMSZ által használt képszolgáltató eszköz egy üvegbura alatt helyezkedik el, melyet kihasználhatunk, ugyanis az esőcseppek bizonyos ideig megmaradnak az üveg felszínén. Ez önmagában lehetőséget biztosít a kiértékelésre, ám csak 10 percenként készül három fényképet tartalmazó sorozatkép, emiatt a pontos időpontokat csak megbecsülni tudjuk. További nehézségeket okoz az, hogy a kamera fókusza 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 az optimális algoritmust, ismernünk kell, hogy milyen tulajdonságokkal rendelkezik egy esőcsepp, mely egy üveglapon – esetünkben egy burán – landol.

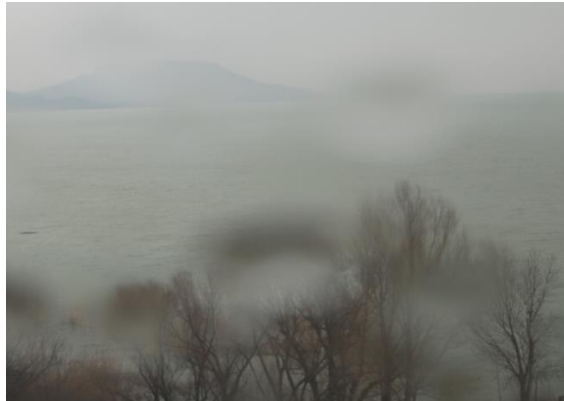
4.2 ESŐCSEPPEK FIZIKAI JELLEMZŐI

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 eltorzult ellipszis felé tendál [12].

Számunkra még ennél is fontosabb jellemzők a következők: [13]

- 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 4.1. ábra).
- A fókusz csökkenésével a cseppek mérete egyre nagyobbak tűnik és kevésbé detektálható az erősen elmosódott hatás miatt.



4.1. ábra - Megfigyelhetők az esőcseppek jellemzői, többek között az, hogy világosabbak, mint a háttér.

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 figyelhetők 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 háttértől, 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 AZ ESŐDETEKTÁLÁS LEHETŐSÉGEI

4.3.1 ESŐVONALAK DETEKTÁLÁSA HISZTOGRAM VIZSGÁLATTAL MOZGÓKÉP SOROZATON

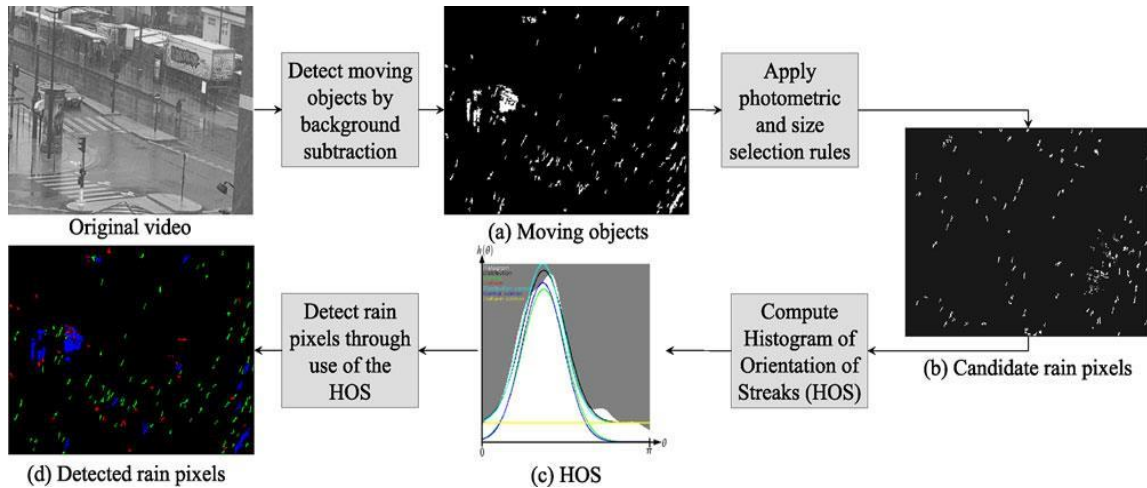
Létezik olyan megközelítése a problémának, amely egy videó képeinek a feldolgozásával kísérli meg az esővonalak felismerését. Az egyes elkészült képeken nem keresünk esőcseppeket, ugyanis a cseppek zuhanási sebessége miatt vonalakként látszanak. Ha két egymást követő képet kivonunk egymásból, akkor többek között ezek a vonalak is látszódní fognak, mint különbség. Következő feladatunk elhagyni azokat az objektumokat, melyek nem az esővonalak által generált különbséget eredményezik [14].

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 (I_{FG}) kivonjuk a háttérmodell (I_{BG}) intenzitásértékét a (4) 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 [14].

$$\Delta I = \text{abs}(I_{FG} - I_{BG}) \geq c \quad (4)$$

Végül az objektumok orientációi alapján eldöntjük, hogy melyik az, amely legjobban jellemzi az esővonalakat. Ehhez az egyes irányokból hisztogramot számolunk, majd ez alapján biztosan megállapíthatóak a keresett vonalak (lásd 4.2. ábra) [14].



4.2. ábra - Eső detektálásának folyamata mozgóképén: (a) mozgó elemek detektálása háttér kivonásával; (b) fényerősségi és méretbeli szabályok alkalmazásával a potenciális területek kiválasztása; (c) a vonalak irányjaiból hisztogramot számolunk; (d) esőpixelek felismerése a hisztogram felhasználásával. [15]

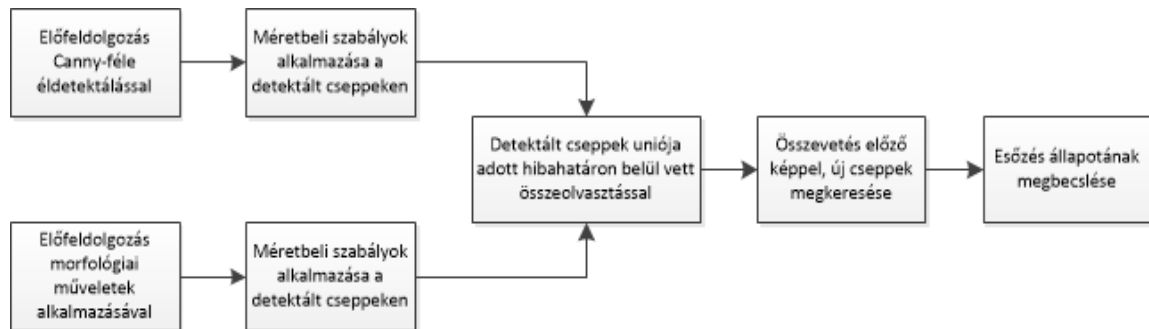
Sajnos esetünkben a mozgókép készítése nem valósítható meg, így az esővonalak detektálásának nincs értelme. A legrövidebb időn belül elkészült képek között is eleredsz vagy megállhat az eső, így a két kép között már valamilyen tartósabb jellemzőket kell keresnünk, erre pont megfelelőek a búrán megmaradt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

4.3.2 ESŐCSEPPEK FELISMERÉSE GÉPJÁRMŰ SZÉLVÉDŐJÉN

Egy másik megközelítés, ami jobban hasonlít a mi problémánkra, 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 becsapódás történt. 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 felismerhetők az egyes elmosódott foltok, melyek a nem fókuszban lévő esőcseppek miatt keletkeztek. [13], [15]



4.3. 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 búrara becsapódó cseppeket úgy detektáljuk, hogy eközben minél kevesebb olyan objektumot ismerjünk fel, amely nem releváns. Valamilyen szegmentációs algoritmusra van szükség, mellyel a különböző foltokat kiválogatjuk. A kiválogatás után pedig meg kell határozni, hogy melyek a keresett objektumok. Ám előbb az input képnek előfeldolgozási folyamaton kell átesnie, aminek eredményeképp a szegmentáció eredményesebb lesz (lásd 4.3. ábra).

A kamerából érkező kép esetén nincs szükségünk feltétlenül a teljes felbontásra, ráadásul a teljes detektálási folyamatot is lelassítaná, ha akkora képmérettel dolgoznánk. 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 szürkeárnyaltos képet készítünk.

Következő feladatunk a zajszűrés. A háttér lehet bármi, így a legfontosabb feladatunk a zavaró pixeleket egy zajszűrő algoritmussal kiszűrni, erre legalkalmasabb tapasztalatunk szerint ebben a feladatrészben a mediánszűrő. Az így létrejött képen még továbbra is nehezen elkülöníthetők az élek. Ahhoz, hogy a szegmentációs algoritmus ne eredményezzen óriási mértékű hibás elemeket, el kell érünk, 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 [16].



4.4. 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, paramétereiktől függően képes az az esőcseppek nagy részét kiválasztani, és még ha a fizikai jellemzőknél leírt okok miatt nem is a teljes kontúrt, de nagy részét megfelelően megtalálja (lásd 4.4. ábra).

4.4.2 ELŐFELDOLGOZÁS MORFOLÓGIAI MŰVELETEK ALKALMAZÁSÁVAL

Ahogy az esőcseppek fizikai jellemzőinél is kiemeltük, az OMSZ által biztosított kamerák végtelen fókuszra vannak állítva. Emiatt és az időjárási viszonyok miatt sajnos előfordulnak olyan esetek, amikor az esőcsepp csak egy elmosódott folt 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 folt-detektálást, ám felkészítjük a rendszerünk egy új algoritmus lefuttatására, amely az elmosódott, nehezen kivehető vízcseppekre van kihegyezve.

A képen, a leírtak szerint valamilyen tömörítést kellene legelőször végrehajtani, majd szürkeárnyaltosítani, végül zajmentesítés szükséges. Tapasztalataink szerint az adaptív simítás [16] itt is jó kimenetet eredményez.

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 4.5. ábra). [17]

4.4.3 ESŐCSEPPEK FELISMERÉSE HIBRID SZEGMENTÁLÓ ALGORITMUSSAL

A folyamat következő lépcsőfoka a szegmentálás, amelyhez rendelkezésre áll két kép. Az első képen már csak annyi dolgunk van, hogy megkeressük a jól kiemelt foltokat, szűrjük őket méretbeli szabályok alkalmazásával, majd rögzítsük a koordinátáit és méreteit a későbbi felhasználás céljából. A második képen viszont további műveletekre van még szükség.



4.5. ábra - Morfológiai műveletekkel végzett esőcsepp detektálás előfeldolgozásának az eredménye.
Balra az eredeti kép, jobbra az eredmény.

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, aminek 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 4.6. ábra). A további pontosítás érdekében a szűrés előtt hasznosnak találtuk minden egyes téglalap méretét 150%-kal megnövelni úgy, hogy középpontjuk

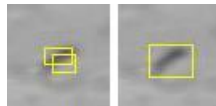
ugyanazon a koordinátán maradjon, így az intenzitásokból fakadó problémák kiküszöbölhetőek.

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 éldetektálás 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.

Először meg kell keresnünk azokat a cseppeket, amelyek ezúttal nem egyforma koordinátán vannak. Kiszűrjük 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.



4.6. 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).

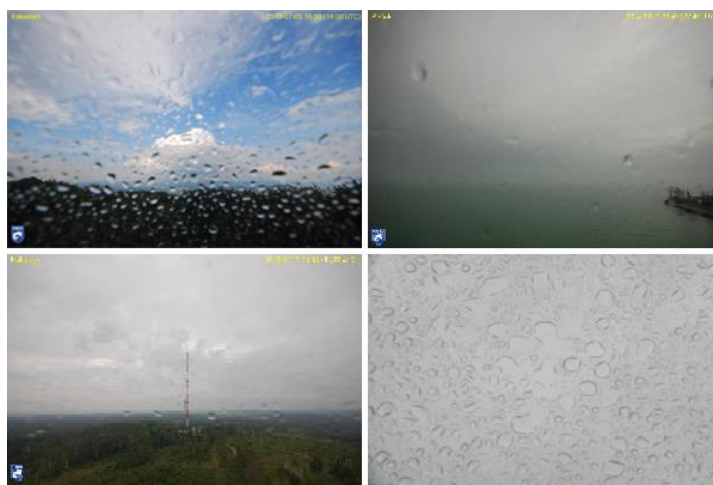
Számunkra elegendő információt ad az, ha tudjuk, hogy az előző képhez képest arányaiban több, vagy kevesebb csepp van. Ha azt tapasztaljuk, hogy több csepp van, mint az időben korábbi képen, akkor feltételezhetjük, hogy azóta esett eső. Ha a cseppek száma jóval kevesebb, mint az első vagy a másodikon 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 4.7. ábra). Egy-egy sorozat a

reggeli óráktól kezdve sötétedésig negyedóránként készített felvételeket tartalmaz. Ezekből egyiken a kamera fölfelé nézett. Az elért eredmények azonban megoszlának.

Az esőcseppek detektálásának a tesztelésére az volt az egyedüli lehetőségünk, hogy mi magunk kézzel jelöltük be a cseppeket, amelyeket a rendszernek fel kellene ismerni. Egy ilyen eredmény kiértékelést tartalmazza a 4.8. ábra. Sajnos elkerülhetetlen volt az, hogy a rendszer fals detektálásokat produkáljon, ugyanis a felhők bizonyos részei pontosan úgy néznek ki, hogy az algoritmus cseppekként ismerje fel őket. A kép a kékestetői állomáson készült. A számszerűsített értékeket a 4. táblázatban láthatjuk.



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



4.8. ábra – Esőcseppek detektálásának a kiértékelése. Zöld négyzetek az elvárt eredményeket jelölik, a sárgák pedig az algoritmus eredményeit. Ez utóbbiakból pirossal lettek jelölve azok a területek, amelyeket a rendszer rosszul ismert fel.

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.

Tesztelt érték neve	Számolt érték
Detektált esőcseppek száma	230
Sikeresen detektált esőcseppek száma	137
Félredetektált cseppek száma	93
Nem detektált cseppek száma	26

4. táblázat – Esőcseppek detektálásának a kiértékelése a kékestetői állomáson.

5 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.

Mozgóképek feldolgozásának számos előnye lehet. Szélirány meghatározásánál már a legkisebb elmozdulás is detektálható és értékelhető. Esőzés vizsgálatánál pedig a pontos időpont is meghatározható, amikor eleredt vagy épp megállt az eső. Két – rövid időn belül – egymást követő kép közötti különbség rámutathat ezúttal nem csak az esőcseppek helyének a megváltozására, hanem magát az esővonalakat is nyomon követheti.

Az Országos Meteorológiai Szolgálat számára a rendszer teljesen egyedülálló és új. 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 költséghatékonnak számít, így más szervezetek, cégek is fel tudják használni ezt a technológiát.

6 ÖSSZEGZÉS

Az Országos Meteorológiai Szolgálat kérésére egy észlelők munkáját támogató rendszert készítettünk, és fejlesztünk folyamatosan. Becsléseiket egy gépi látó rendszer segítségével pontosítjuk, mégpedig a következő területeken: felhő borultságának a vizsgálata, felhő típusainak a megállapítása, magassági szélirány meghatározása végül az esőzés kezdetének és befejeződésének megbecslése. Különböző kényszerfeltételek betartása mellett kellett az algoritmusainkat optimalizálni. Ilyen feltételek például a viszonylag ritka – 10 percenkénti – fénykép készítése, a már meglévő webkamerák felhasználása, melyek többnyire a horizontot kémlelik, adott esetben az eget figyelik. Méréseink nem tökéletesek, ám ennek ellenére az észlelők munkáját támogatja, pontosítja.

A borultság vizsgálata és a típus meghatározása során 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, azaz 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.

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). Esős képek esetében a tesztek alapján az ilyen képek 85.71%-a helyesen lett detektálva.

A szélirány meghatározása során értelem szerűen a felhők elmozdulásának az irányára voltunk kíváncsiak. Az teszteket az OMSZ által biztosított 1000 felvételtől álló sorozatképen végeztük el, melyek között található volt Cumulus és Stratus típusú felhő – lehetővé téve a változatos színű és a homogén képeken való tesztelést – valamint esőcseppes képek is. Az elvégzett teszteken az algoritmus változatos színű felhőkkel jól, míg Stratus típusúakkal rosszabbul szerepelt. Előbbinél az esetek többségében helyesen sikerült a szélirány becslése, a legnagyobb hiba fél égtáj eltérés

volt. Homogén felületen azonban az egyértelmű képpontok hiánya miatt nem lehet pontos becslést tenni. A sorozatokon végzett tesztel eredményeit javítottuk a képek számának növelésével, így a teszteket elvégeztük tíz képes sorozatokon is. Így Cumulus típusú felhők esetén sikerült elkerülnünk az eddigi minimális eltéréseket is. Az eredmény homogén felületen is javult, de messze elmaradt a változatos felhőzet mellett. A megfelelő eredmények érdekében felhasználtuk a borultság vizsgálat eredményeit, és amennyiben Stratus típusú felhőzetünk volt, "Nem meghatározható"-nak tekintettük az irányt. Ugyanezt tettük esőzés esetén is, mivel a búrán lévő esőcseppek miatt a helyes detektálás lehetetlen, ráadásul csapadékhoz a legtöbb esetben 8 oktás homogén felhőzet tartozik.

Esőzés vizsgálatának során igyekeztünk minél pontosabb darabszámot adni a képen látható esőcseppekre. Minél pontosabban tudjuk, hogy hol láthatóak a cseppek, annál pontosabban tudjuk eldönteni, hogy a két időpont között, ahol kép készült, változott-e az esőzés állapota. Az OMSZ számunkra biztosított képgyűjteményben 4 olyan képsorozat található, amelyeken van értelme esődetektálást tesztelni. Egy-egy sorozat a reggeli óráktól kezdve sötétedésig negyedóránként készített felvételeket tartalmaz. Ezekből egyiken a kamera fölfelé nézett. Az elért eredmények azonban megoszlottak. Sajnos elkerülhetetlen volt az, hogy a rendszer fals detektálásokat produkáljon, ugyanis a felhők bizonyos részei pontosan úgy néznek ki, hogy az algoritmus cseppekként ismerje fel őket. 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 síófoki 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, de az észlelők számára a kezdés időpontja a lényeges, a befejezés csak plusz hasznos információ.

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.

7 IRODALOMJEGYZÉK

- [1] Országos Meteorológiai Szolgálat, *Gépi látó rendszerünk fejlesztésének a támogatására elkészült írásos specifikáció és képgyűjtemény*, Budapest, 2013.
- [2] Met Office, Cloud types for observers, Reading the sky, 2006.
- [3] 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., March 2006..
- [4] F. Samopa and A. Asano, "Hybrid Image Thresholding Method using Edge Detection," *IJCSNS Int. J. of Comput. Sci. and Netw. Secur.*, vol. 9, no. 4, pp. 292-299., April 2009.
- [5] L. Ding and A. Goshtasby, "On the Canny edge detector," *Pattern Recognition*, vol. 34, pp. 721-725, 2001.
- [6] 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, pp. 723-727, March 1991..
- [7] J. V. Miss Hetal és B. Prof. Astah, „A Review on Otsu Image Segmentation Algorithm,” *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 2, no. 2, pp. 387-389, 2013.
- [8] B. K. Choudhary, N. K. Sinha and P. Shanker, "Pyramid Method in Image Processing," *Journal of Information Systems and Communication*, vol. 3, pp. 269-273., 2012.
- [9] Z. Vámosy, Á. Tóth and P. Hirschberg, "PAL Based Localization Using Pyramidal Lucas-Kanade Feature Tracker," in *IEEE Proceedings of the 2nd Serbian-Hungarian Joint Symposium on Intelligent Systems*, Subotica, Szerbia, 2004, pp. 223-231..
- [10] D. Fleet and Y. Weiss, "Optical Flow Estimation," in *Handbook of Mathematical Models in Computer Vision*, Springer, 2005, pp. 239-257.
- [11] R. Zhang, „Automatic Computation of a Homography by RANSAC Algorithm,” *ECE661 Computer Vision Homework 4*, pp. 1-33.

- [12] K. Garg and S. K. Nayar, "Vision and Rain," *International Journal of Computer Vision*, vol. 75, no. 1, pp. 3-27, 2007.
- [13] F. Nashashibi, R. de Charette és 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.
- [14] 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.
- [15] H. Kurihata, T. Takahashi, I. Ide, Y. Mekada, H. Murase, Y. Tamatsu és T. Miyahara, „Rainy weather recognition from in-vehicle camera images for driver assistance,” *IEEE Proceedings. Intell. Veh. Symp. 2005*, pp. 205-210, 2005.
- [16] C. Kervrann, „An Adaptive Window Approach for Image Smoothing and Structures Preserving,” *Processing*, pp. 132-144, 2004.
- [17] A. Cord és G. Nicolas, „Detecting Unfocused Raindrops In-Vehicle Multipurpose Cameras,” *IEEE Robot. Autom. Mag.*, vol. 21, no. 1, pp. 49-56, 2014.

8 ÁBRAJEGYZÉK

1.1. ábra - A rendszer fejlesztése során használt NIKON D50 fényképezőgép, melyhez hasonlóak jelenleg az ország több mint 9 pontján készítenek rendszeresen képeket az OMSZ számára [1].	5
1.2. ábra - Az OMSZ által használt kamerák szerkezeti felépítése [1].	6
2.1. ábra - Égbolt borultságának meghatározása becsléssel, képzeletbeli rács használatával. A kép alapján becsült érték 3 okta [2].	7
2.2. ábra – Cumulus típusú felhő, melynél jól látható a felfelé való terjeszkedés, mely egy határ után elterül. A folyamat egyetlen kép alapján nem detektálható, felfelé néző kamera esetén a felső rész nem is látható [2].	8
2.3. ábra – Halo jelenség különböző megnyilvánulásai vékony felhőzet esetén. [2]	9
2.4. ábra - A két felhőtípus, melyeket gépi látó rendszerrel meg lehet különböztetni. Balról a Stratus, jobbról Cumulus látható.	11
2.5. ábra - A hibrid küszöbölést és éldetektálást alkalmazó algoritmus hibás működése teljes eget borító, változatos színű felhők esetén. Látható, hogy ilyen esetben is határoz meg küszöbértéket, ami hibás küszöbölést eredményez.	16
2.6. ábra - RGB kocka a kék, zöld és vörös értékek ábrázolására [3].	16
2.7. ábra - Az IHS színtér ábrázolása.(a) Színháromszög, melynek pontjai a zöld, vörös és kék értékek. (b) Árnyalat, telítettség és intenzitás kombinációja háromdimenziós színtérben [3].	17
2.8. ábra – (a) Eredeti kép, amin a piros egyenes jelzi a vizsgált tartományt. (b) Szaturációs értékek az eredeti képen található vonal egyes részein. Jól megfigyelhető az ég és felhő értékei közti különbség [3].	18
2.9. ábra – Osztályok a pixelek besorolására. Class 1: tiszta ég; Class 2: nem meghatározott; Class 3: felhő [3].	19
2.10. ábra – Borultság vizsgálat folyamata	20
2.11. ábra – Az első képen az elmosás nélküli szaturációs térkép látható, míg a második az AForge Blur algoritmusát, a harmadik pedig a Gauss módszert alkalmazza....	21
2.12. ábra – Eredeti kép, szaturációs kép, küszöbölt kép – kék: ég, fehér: felhő, zöld: nem besorolt.....	22
2.13. ábra – Felhőtlen ég esetén zajt tapasztalhatunk, melynek kezelésére külön figyelmet kell fordítanunk.	22
2.14. ábra – Felhő típus detektálásának folyamata	24
2.15. ábra – A rendszer hasonló, esőcseppekkel borított képek esetén is sikeresen működött. Ez esetben az eredmény 8 oktás Stratus felhőzet.	25
2.16. ábra – Látható, hogy míg horizontálisan elhelyezett kamera képén a naplemente jelentősen elszínezi a felhőzetet, azonos időpontban elkészített képen, felfelé néző kamera esetén ez a jelenség csak minimálismértékben tapasztalható.	26
3.1. ábra – A szélirány meghatározásának a lépései.	29

3.2. ábra – Sarokpontok detektálása felhős képen előfeldolgozó algoritmus futtatása előtt (balra) és után (jobbra).....	31
3.3. ábra – Sarokpont detektorok alkalmazása felhős képen. Balra a Harris-féle sarokpont detektor, jobbra a minimális sajátértékkel számolt módszer.	32
3.4. ábra – Az optikai áramlás eredménye korrekciók nélkül (balra), majd megfelelő korrigálások elvégzése után (jobbra) – pirossal az elmozdulások.	33
4.1. ábra - Megfigyelhetők az esőcseppek jellemzői, többek között az, hogy világosabbak, mint a háttér.	36
4.2. ábra - Eső detektálásának folyamata mozgóképen: (a) mozgó elemek detektálása háttér kivonásával; (b) fényerősségi és méretbeli szabályok alkalmazásával a potenciális területek kiválasztása; (c) a vonalak irányából hisztogramot számolunk;(d) esőpixelek felismerése a hisztogram felhasználásával. [15].....	37
4.3. ábra - Esődetektálás teljes folyamata.....	38
4.4. ábra - Éldetektálással végzett esőcsepp detektálás előfeldolgozásának az eredménye.	39
4.5. ábra - Morfológiai műveletekkel végzett esőcsepp detektálás előfeldolgozásának az eredménye. Balra az eredeti kép, jobbra az eredmény.	40
4.6. 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).	41
4.7. ábra – Algoritmusunk tesztelése különböző helyszíneken. Sorrendben balról jobbra, fentről lefelé: Kékestető, Siófok, Kab-hegy, ismeretlen helyszínen fölfelé néző kamera.	42
4.8. ábra – Esőcseppek detektálásának a kiértékelése. Zöld négyzetek az elvárt eredményeket jelölik, a sárgák pedig az algoritmus eredményeit. Ez utóbbiakból pirossal lettek jelölve azok a területek, amelyeket a rendszer rosszul ismert fel....	42
10.1. ábra – Egyszerű detektálás eredménye vertikális helyzetű képek alapján. Balra a részeredmények, míg jobbra a végeredmények láthatók.	58
10.2. ábra – Több kép megnyitása az „Open vertical images...” vagy az „Open horizontal images...” menüpont kiválasztása után.	60
10.3. ábra – IImageProcessService és az általa használt SubresultAvailableEventHandler delegált felépítése.....	61
10.4. ábra – ISubprocessService felépítése.....	61
10.5. ábra – A képek egyszerű és gyors megnyitására létrehozott kisegítő osztály szerkezete.....	62
10.6. ábra – Felhő detektálás küszöbértékeinek manuális beállítása három eltérő típusú felhő alapján. A fenti sorban az eredeti képek, míg az alsóban a küszöbölt képek láthatók.....	63
10.7. ábra – A felhők típusát és az égbolt borultságát meghatározó modul felépítése...	64
10.8. ábra – A felhők típusának valamint az égbolt borultságának meghatározását végző metódusok felépítése. A fenti sorban elhelyezkedő osztályok megvalósítják az	

ISubProcessService interfészt, míg az alul lévők esetében erre nincs lehetőség, mivel nem végeznek a képen módosítást.....	65
10.9. ábra – A felhők mozgásának meghatározására szolgáló alap modul szerkezete... 66	66
10.10. ábra – A felhők mozgását detektáló osztály felépítése. Az ISubProcessService interfész megvalósítására nincs lehetőség, mivel nem végez módosítást a képen. 66	66
10.11. ábra - Az esőcseppek detektálását egy IRaindropDetector interfész implementációja végzi.	67
10.12. ábra - Az esőcseppek detektálásához szükséges paramétereket egy IRaindropDetectionContext interfészen keresztül adjuk át. Ugyanezen az interfészen keresztül utazzatjuk a folyamat során keletkezett paramétereket és a végeredményt is.	67
10.13. ábra - A felhasználó által tesztelés céljából bejelölt esőcseppeket a képen az IRainDetectionTester interfész implementációja veti össze. Így értékelhetőek az egyes algoritmusok.	69
10.14. ábra - A felhasználó által megadott esőcseppek és az algoritmus által detektált cseppek összevetésének az eredményét egy IRaindropDetectionTestResult interfészen keresztül adjuk vissza.	69
10.15. ábra – A borultságot és felhő típusát figyelő modul felhasználói felülete grafikus megjelenítés esetén.	70
10.16. ábra – Grafikus rendszerünk működése esőcseppek detektálása közben. Látható, hogyan milyen útvonalat járt be az algoritmus, ameddig a végső érték meg nem született.	71
10.17. ábra – Funkciók végrehajtása egyenként grafikus esődetektálás esetén.....	72

9 MELLÉKLETEK

Latin név	Magyar név	Szint
Cirrus	pehelyfelhő	Magas
Cirrocumulus	báránfelhő	Magas
Cirrostratus	fátyolfelhő	Magas
Alto cumulus	párnafelhő	Középmagas
Altostratus	lepelfelhő	Középmagas
Stratocumulus	gomolyos rétegfelhő	Alacsony
Stratus	rétegfelhő	Alacsony
Cumulus	gomolyfelhő	Alacsony
Nimbostratus	esőrétegfelhő	Középmagas
Cumulonimbus	zivatarfelhő	Több szintet átfog

1. melléklet - Felhők típusai az [1] és [2] művek alapján.

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. melléklet – Szakirodalomban meghatározott küszöbértékek a felhődetektálás során alkalmazott osztályoknál.

(Fa - felhő alsó, Ff - felhő felső, Ea - ég alsó, Ef - Ég felső korlátja) [3]

Észlelők által meghatározott típus	Program által meghatározott típus	Észlelők által meghatározott borultság	Program által meghatározott borultság
Cumulus	Cumulus	5	2
Cumulus	Cumulus	5	1
Cumulus	Clear	5	0
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Stratus	7	8
Stratus	Cumulus	8	7
Stratus	Stratus	8	7
Stratus	Stratus	8	8
Stratus	Stratus	8	8
Stratus	Stratus	8	7
Stratus	Stratus	8	7
Stratus	Stratus	8	7
Stratus	Stratus	8	7
Stratus	Stratus	8	8
Stratus	Cumulus	8	7
Stratus	Stratus	8	7
Stratus	Stratus	8	7
Stratus	Stratus	8	7
Stratus	Stratus	8	7
Stratus	Cumulus	8	7
Stratus	Cumulus	8	7
Cumulus	Stratus	7	7
Cumulus	Cumulus	6	7
Cumulus	Cumulus	1	7

Cumulus	Stratus	2	2
Cumulus	Stratus	5	5
Stratus	Cumulus	7	6
Stratus	Cumulus	7	7
Stratus	Stratus	8	8
Stratus	Stratus	8	8
Stratus	Stratus	8	8
Stratus	Stratus	8	8
Stratus	Cumulus	8	7

3. melléklet – Borultság és felhőtípus vizsgálata különálló képek alapján.

Észlelők által meghatározott típus	Program által meghatározott típus	Észlelők által meghatározott borultság	Program által meghatározott borultság
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Stratus	7	8
Cumulus	Stratus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Stratus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	6

Cumulus	Stratus	7	6
Cumulus	Stratus	7	0
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Cumulus	7	7
Cumulus	Stratus	7	8
Cumulus	Stratus	7	8
Cumulus	Stratus	7	8

4. melléklet – Borultság és felhő típus vizsgálata sorozatképek alapján.

10 A PROGRAMRENDSZER BEMUTATÁSA

Projektünk fejlesztése Visual Studio 2013 fejlesztőkörnyezetben C# nyelven zajlott, a .NET keretrendszert felhasználva. Mivel az alkalmazás felületén nem volt semmilyen speciális elvárás, a Windows Forms sémát választottuk a program alapjául. A rendszer több különálló modulból áll, amik folyamatosan fejlődhetnek és változhatnak, így célszerűnek tűnt a fejlesztést interfészek definiálásával kezdeni, hogy egységessé tegyük az egyes egységek szerkezetét, és megteremtsük a kapcsolatot a modulok között. Hogy a fejlesztést gyorsítsuk, bizonyos képfeldolgozást végző algoritmusokat nem mi készítettünk el, hanem külső könyvtárakat használtunk. Ilyenek az AForge és az Accord.NET, melyeket a Visual Studio-ba beépített NuGet Package Manager-rel telepítettünk, ami lehetővé teszi ezek egyszerű letöltését és frissítését. Hogy a rendszer fejlődését könnyen nyomon követhessük, megoszthassuk azt, és a különböző modulokon a többi rész befolyásolása nélkül dolgozhassunk, így a tárolásra Git-et használtunk, ami a fent említett lehetőségek mellett verziókezelésre is lehetőséget ad.

10.1 A RENDSZER MODULJAI

A programot különböző modulokra osztottuk. Ennek célja a rendszer dinamikusabbá tétele, átláthatóságának javítása. Minden modul egy-egy különálló könyvtárat jelöl, amiket új fejlesztések esetén könnyedén kicserélhetünk. A rendszer moduljai a következők:

- fő modul
- interfész modul
- almodulok

A fő modul tartalmazza a kinyert információval való műveletvégzéshez tartozó logikát. Ide tartozik a felhasználói felület megjelenítése, kezelése, adatok exportálása, továbbítása, beállítások, valamint a konfigurálás lehetősége. A felsorolt funkciók közül nem került mindegyik megvalósításra, mivel hogy a kinyert információval milyen műveletet végezzünk automatikusan fog zajlani, és különböző alkalmazási területeken más-más lehet.

Hogy az almoduljainkat dinamikusan változtathassuk, és egyszerűen bővíthessük a programunkat, egy köztes réteget is létrehoztunk, ami megteremti a kapcsolatot a főmodul és almodulok között. Ez az interfész modul, aminek betartásával a főmodul az almodul pontos ismerete nélkül is képes információt fogadni. Almoduljaink jelenleg a felhők mozgásának, típusának-, az égbolt borultságának meghatározására szolgáló-, és az esőcseppek detektálásáért felelős modul.

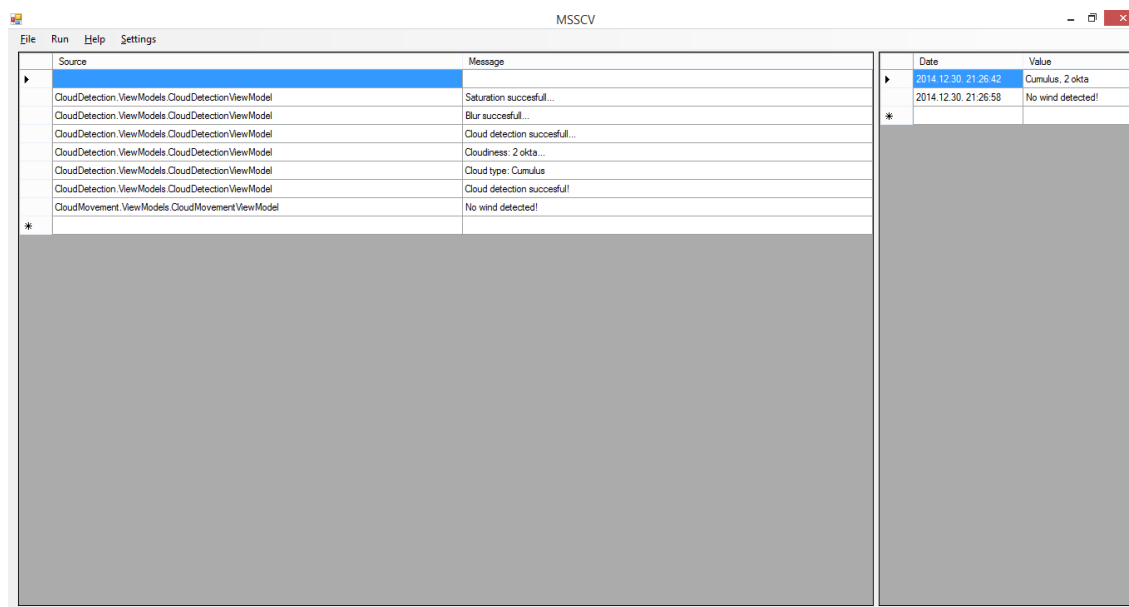
10.2 FŐ MODUL

Az alkalmazás legfelső rétegének feladata az almodulok által kiszámított adatokkal való műveletvégzés, valamint a felhasználói felület megjelenítése, vezérlése. Az információ interfészeken keresztül érkezik, így a főmodul a nélkül képes adatot fogadni, hogy a küldő osztály működését pontosan ismerné. A beérkezett adatokat már az igényeknek megfelelően kezelhetjük. Exportálhatjuk szöveges dokumentumban, vagy interneten keresztül továbbíthatjuk más állomások számára.

10.2.1 FELHASZNÁLÓI FELÜLET

Jelenleg az adatok táblázatban való megjelenítésére van lehetőség. Ennek oka, hogy az OMSZ által használt rendszerek jelenleg speciális routerek, melyekhez nem tartozik állandó kijelző, így a képek megjelenítésére nincs szükség. A program nyelvének az angolt választottuk.

Az elsődleges ablak vertikálisan két részre van osztva. Baloldalon a végrehajtás folyamatáról kaphatunk visszajelzést, míg jobb oldalon a végeredmények lettek megjelenítve úgynevezett DataGridView elem segítségével (lásd 10.1. ábra). Ilyen módon habár képekkel nem illusztráljuk a program működését, folyamatos tájékoztatást kaphatunk arról, hogy a program fut és működik. Minden részfeladat esetén láthatjuk, hogy melyik modultól származik, valamint valamilyen szöveges visszajelzést a folyamat eredményéről. A végeredmények esetében csak a számított érték látható a lehető legtömörebb, de még egyértelmű formátumban megadva, valamint a számítás időpontja.



The screenshot shows the MSSCV application window. The left pane displays a log of events with columns 'Source' and 'Message'. The right pane displays a summary table with columns 'Date' and 'Value'.

Source	Message
CloudDetection.ViewModel.CloudDetectionViewModel	Saturation successful...
CloudDetection.ViewModel.CloudDetectionViewModel	Blur successful...
CloudDetection.ViewModel.CloudDetectionViewModel	Cloud detection successful...
CloudDetection.ViewModel.CloudDetectionViewModel	Cloudiness: 2 okta...
CloudDetection.ViewModel.CloudDetectionViewModel	Cloud type: Cumulus
CloudDetection.ViewModel.CloudDetectionViewModel	Cloud detection successful!
CloudMovement.ViewModel.CloudMovementViewModel	No wind detected!

Date	Value
2014.12.30. 21:26:42	Cumulus, 2 okta
2014.12.30. 21:26:58	No wind detected!

10.1. ábra – Egyszerű detektálás eredménye vertikális helyzetű képek alapján. Balra a részeredmények, míg jobbra a végeredmények láthatók.

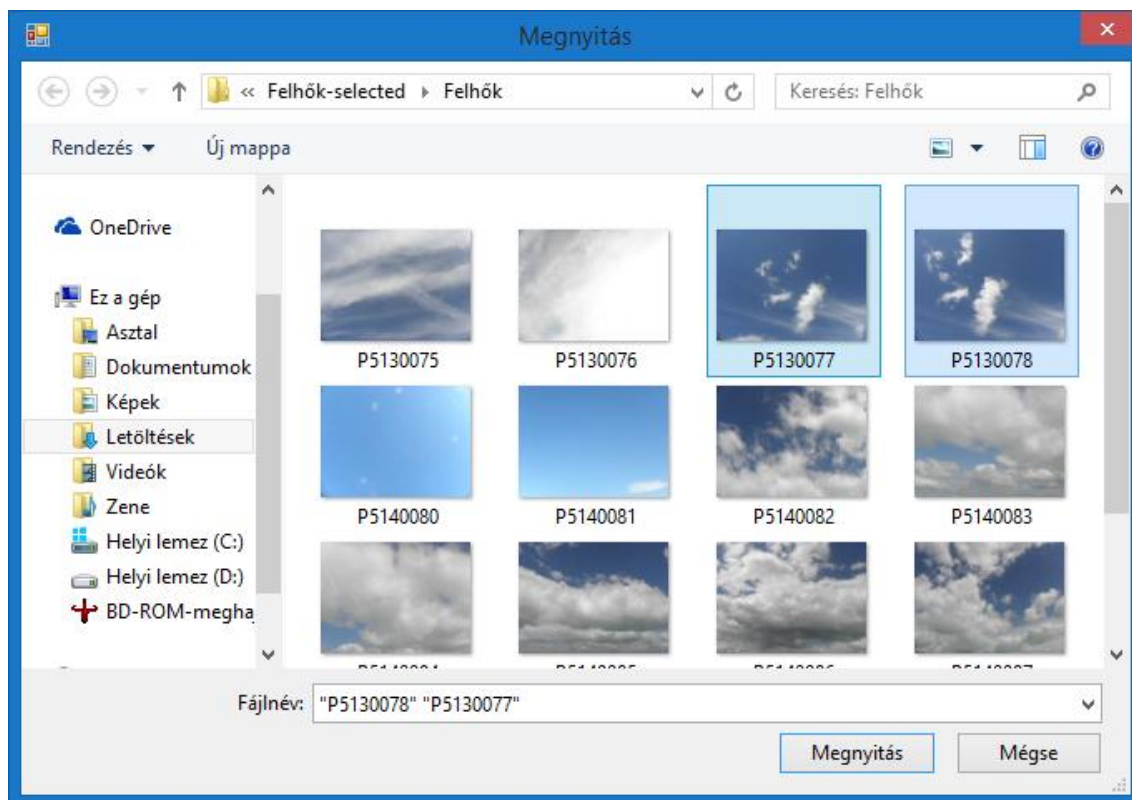
A táblázatok fölött egy menü sáv lett elhelyezve, amivel a programot vezérelhetjük, elvégezhetjük a beállításokat. Éles környezetben ez automatikus folyamat lesz, azonban a képek betöltése még a felhasználó által történik. Első lépésként a „File...” menüpont „Open vertical images...” és „Open horizontal images...” parancsát érdemes használnunk, előbbi a felfelé néző kamerából származó képeket, míg utóbbi a vízszintesen készületeket jelenti (lásd 10.2. ábra). Ezek segítségével választhatjuk ki a képeket, amiken a későbbi detektálást végezni szeretnénk. Mindkét esetben több kép megnyitására van lehetőség, egyes funkciók működéséhez legalább két kép megnyitása szükséges is. Hogy melyik képeket melyik modulok használják, azt már a program dönti el, a detektálás a képeken egyszerre történik, nincs lehetőség modul kikapcsolására. A futtatást a „Run” gombra kattintva kezdetjük el, ami után az információk egyből elkezdenek megjelenni a táblázatokban. Mivel az eredmények beérkezéséről a program értesül, így lehetőség van arra is, hogy az egyes almodulok felhasználják a már meglévő, máshonnan származó értékeket.

A „Settings...” menüpont a rendszer konfigurálására szolgál, funkcióit később részletesen ismertetjük. Az itt kiválasztott elemek mind külön ablakban nyílnak meg. Új folyamatot nem lehet indítani, amíg a konfigurálást be nem fejeztük, az ablakot be kell zárni.

A „Help...” menüpont alatt található gombra kattintva („About MSSCV...”) egy felugró ablakban a programmal kapcsolatos információk jelennek meg.

10.3 INTERFÉSZEK

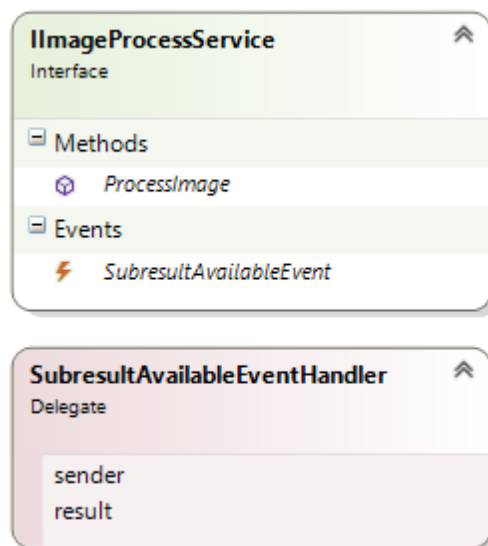
A program általános esetben az észlelőktől távol, egy speciális router-re telepített operációs rendszeren fog futni, amihez nincs monitor csatlakoztatva. Ebből kiindulva programunk nem a grafikus elemek, hanem a kinyert információ kezelésére lett kialakítva. A modulok közötti egységet a lehető legegyszerűbb módon definiált interfészekkel próbáltuk megteremteni. Ezek közül nem mindegyik betartása kötelező, egyesek csak a helyes megvalósítást segítik. Természetesen a kiemelteken kívül az egyes részeknek is lehetnek saját interfészei, de most csak a teljes programra kiterjedőkkel foglalkozunk.



10.2. ábra – Több kép megnyitása az „Open vertical images...” vagy az „Open horizontal images...” menüpont kiválasztása után.

10.3.1 IMAGE PROCESS SERVICE INTERFÉSZ

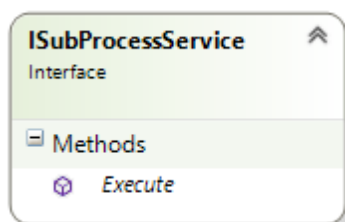
A kapcsolat a fő modul és az almodulok között ennek megfelelően zajlik, alkalmazása minden esetben kötelező. A korábban leírtak szerint csak a kinyert információ átadása a lényeges. Ennek megfelelően az interfész egy metódust tartalmaz, ami képek listáját várja paraméterként, és egyetlen szöveg a visszatérési értéke, ami a végeredményt tartalmazza, lehetőleg minél egyszerűbben és érthetőbben összefoglalva (lásd 10.3. ábra). Azonban a teljes feldolgozás hosszú időt vehet igénybe, és így a felhasználó semmilyen visszajelzést nem kap annak állapotáról. Ennek kiküszöbölésére az interfész egy eseményt is definiál, amiből a küldő osztály, és egy szöveg nyerhető ki. Ez akkor lesz meghívva az almodulokban, ha egy folyamat véget ért, és így annak eredményéről is tájékoztathatjuk a fő modult, láthatóvá válik, hogy folyik a feldolgozás, illetve annak eredményéről is információt kaphatunk. Betartása minden almodul esetén kötelező, mivel a főmodulunk csak ilyen interfészekon keresztül kommunikál.



10.3. ábra – IImageProcessService és az általa használt SubresultAvailableEventHandler delegált felépítése.

10.3.2 SUBPROCESS SERVICE INTERFÉSZ

Betartása nem kötelező, ugyanis nem szolgál információval a fő modul számára, azonban minden olyan folyamatnál ajánlott alkalmazni, ami egy képen végez valamilyen módosítást.

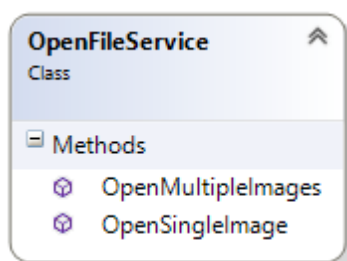


10.4. ábra – ISubprocessService felépítése

Mivel a képek nagyméretűek, így helytelen kezelésük esetén a memória hamar megtelik, ami "OutOfMemoryException"-t eredményez. Hogy ezt elkerüljük, célszerű minden esetben az eredeti képen végezni a módosításokat, nem újat készíteni. Azokban az esetekben, amikor az eredeti kép megtartása is szükséges bizonyos ideig, másolhatjuk azt, majd használat után töröljük a memóriából. Ennek a szisztémának a követésére az interfész egyetlen metódust tartalmaz, ami egy képet vár paraméterként, és nincs visszatérési értéke (lásd 10.4. ábra). Ennek hiánya miatt kénytelenek leszünk a műveleteket a paraméterként kapott képen végrehajtani, és így az algoritmusok több százszori futtatása esetén sem kaphatunk hibaüzenetet.

10.4 KISEGÍTŐ OSZTÁLYOK

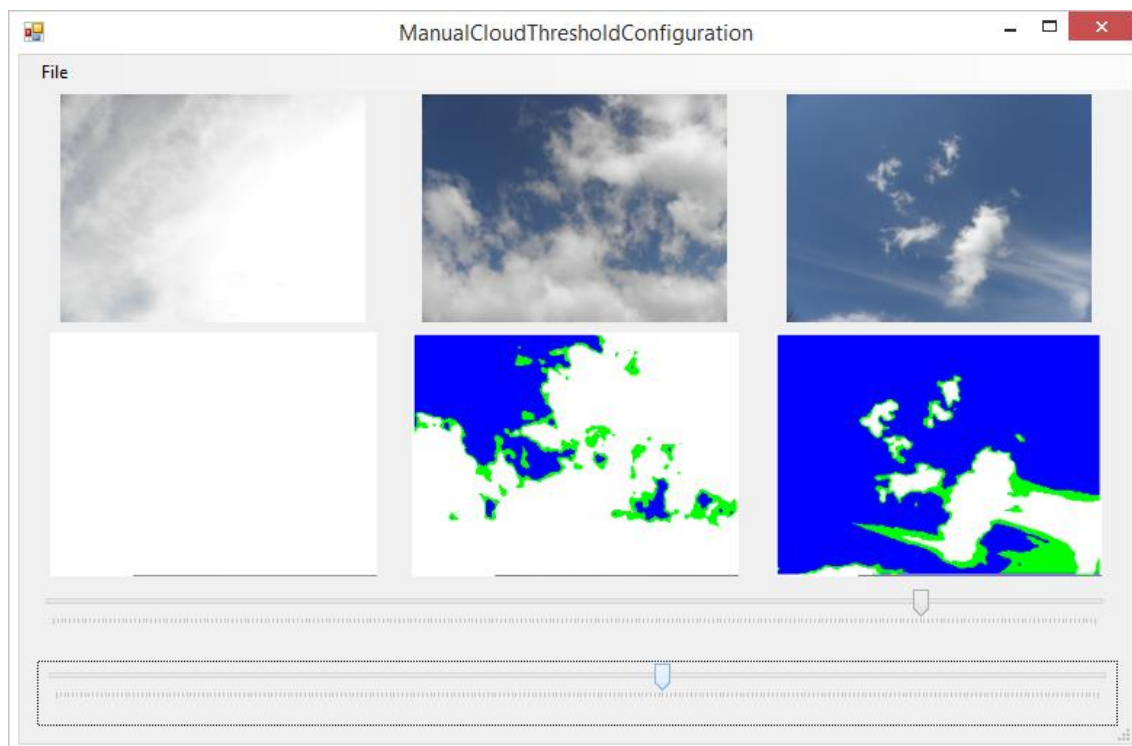
A későbbi munka segítésére az olyan funkciókat, melyeket a programban gyakran használunk, külön osztályban implementáltuk. Ilyen módon elegendő az algoritmust egyszer megírni, és a továbbiakban azt egyetlen metódushívással használhatjuk, hosszas gépelés helyett. Így gyorsabb a munka, valamint a hiba behatárolása és javítása is egyszerűbbé válik. Jelenleg az egyetlen ilyen a programban megvalósított osztály a képek megnyitására szolgál. Ezt azért tartottuk szükségesnek ilyen módon elkülöníteni, mivel ez lesz a program egyik leggyakrabban használt funkciója, aminek megvalósítása több beállítást is igényelhet. Ilyen például a megnyitni kívánt fájlok típusa, visszaálljon-e minden megnyitáskor a legutóbb látogatott könyvtárra, valamint a legfontosabb beállítás, hogy egy vagy több fájlt szeretnénk-e megnyitni. Ezek megvalósítására az osztályunk két statikus metódust tartalmaz, melyek így az osztály példányosítása nélkül is meghívhatók (lásd 10.5. ábra). Az első egyetlen-, míg a második több kép megnyitására szolgál.



10.5. ábra – A képek egyszerű és gyors megnyitására létrehozott kisegítő osztály szerkezete.

10.5 FELHŐ DETEKTÁLÓ MODUL

Feladata az égbolt figyelése, és az alapján az aktuális borultság, felhő típus és magassági szélirány megállapítása. Ezek elvégzéséhez csak a felfelé néző kamera képe szükséges. A teljes feldolgozás hat lépésből áll, és a felhasználót mindegyik végén tájékoztatjuk annak eredményéről (lásd 10.7. ábra). Mivel a tesztek alapján, a sorozatképeken végzett vizsgálatok esetében kaptunk pontosabb eredményt, így több kép paraméterként való megadását is támogatja, de egyen is eredményesen működik. A modul minden eleme, ami képen végez műveleteket, megvalósítja a fent említett „ISubprocessService” interfészt. A feladatok elvégzésével a kiszámított felhő típust és borultságot vesszővel elválasztva juttatja vissza a főmodulhoz (lásd 10.8. ábra).

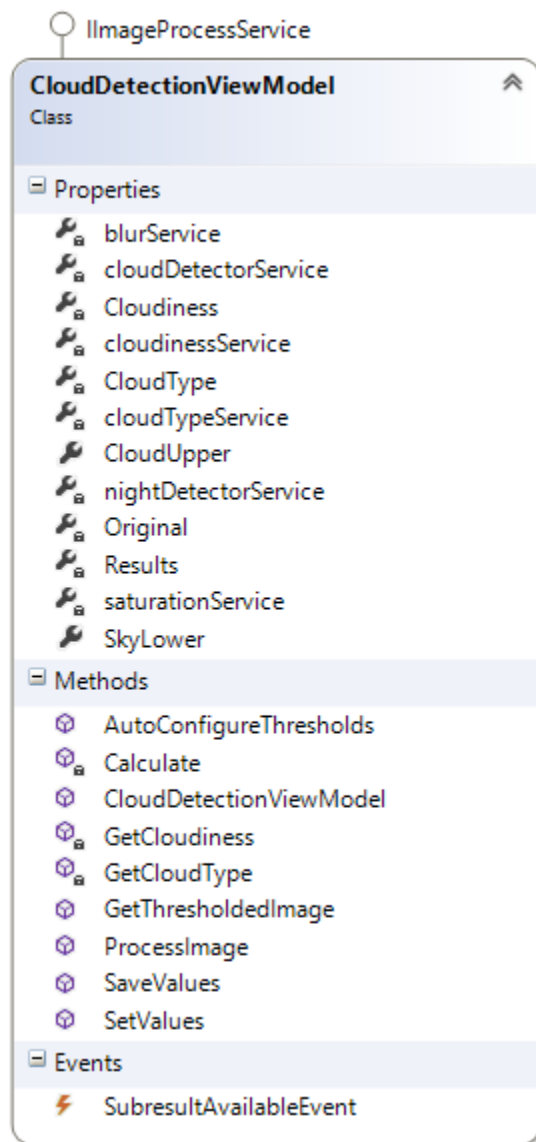


10.6. ábra – Felhő detektálás küszöbértékeinek manuális beállítása három eltérő típusú felhő alapján. A fenti sorban az eredeti képek, míg az alsóban a küszöbölt képek láthatók.

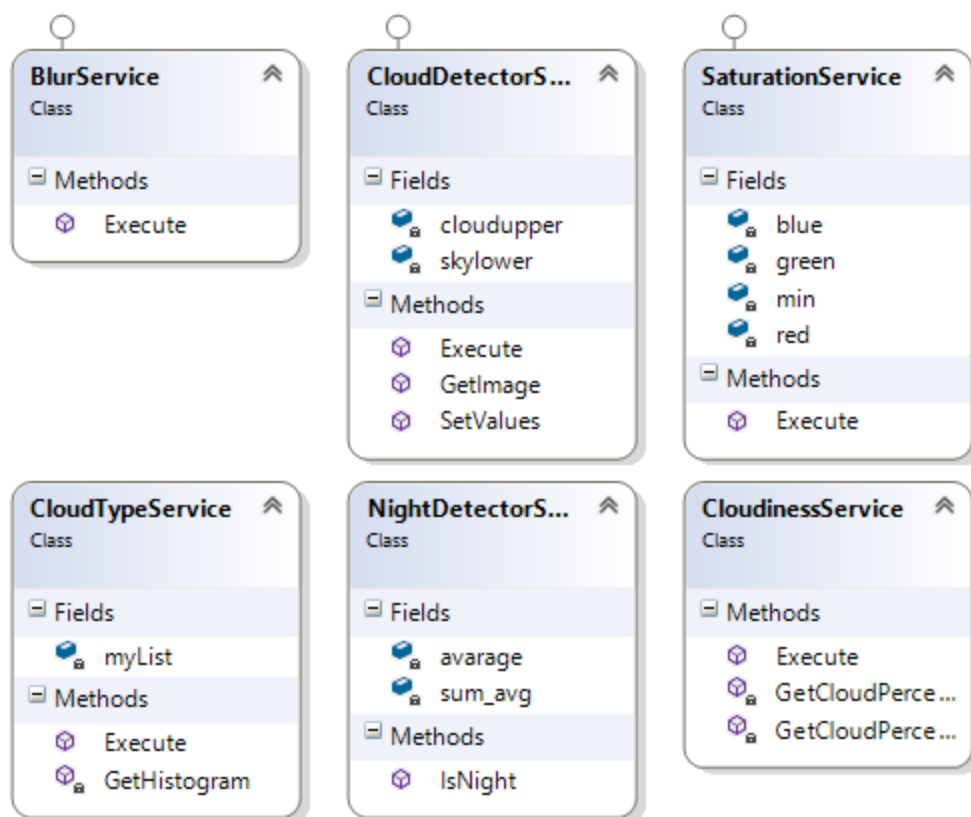
A „Settings...” menüpont alatt található „Cloud thresholds” almenüpont tartozik a modulhoz. Ez alatt másik két opció található: „Automatic” és „Manual”. Utóbbit kiválasztva a megnyílt ablakban lesz lehetőségünk a borultság vizsgálatnál használt küszöbértékek manuális. A „File” menüpont három gombot tartalmaz, melyek segítségével különböző típusú képeket nyithatunk meg. Célszerű két nagyban eltérő és egy köztes állapotú felhőt választani. A megnyitást követően a képek horizontálisan lesznek elhelyezve az ablak felső sorában, míg alattuk a küszöbölés végeredményeként létrejött képek láthatók (lásd 10.6. ábra). Az ablak alján található csúszka segítségével valós időben módosíthatjuk a küszöbértéket. Az alsó a felhők alsó határát, míg a felső az ég alsó határát módosítja. Célszerű a fenti csúszka beállításával kezdeni, majd az alsót hozzá igazítani. Az egyes rétegeket a jól elkülöníthetőség érdekében a korábban már meghatározott színek jelölik. A mentés az ablak bezárásával automatikusan történik.

Ezzel a módszerrel pontosan beállíthatjuk a programot, azonban időigényes. Gyorsabb lehetőséget biztosít az automatikus konfigurálás, amivel egyetlen gombnyomással állíthatjuk be a küszöbértékeket. Ezt az „Automatic” gomb kiválasztásával tehetjük meg. Azonban mielőtt rákattintanánk, meg kell nyitnunk egy képet az „Open vertical images...” gomb segítségével. Több kép megnyitása esetén az első alapján végzi a beállítást. Ha a kapott értékekkel mégsem lennénk megelégedve,

utána még van lehetőség manuálisan pontosítani, így célszerű mindig az automatikus beállítással kezdeni.



10.7. ábra – A felhők típusát és az égbolt borultságát meghatározó modul felépítése.



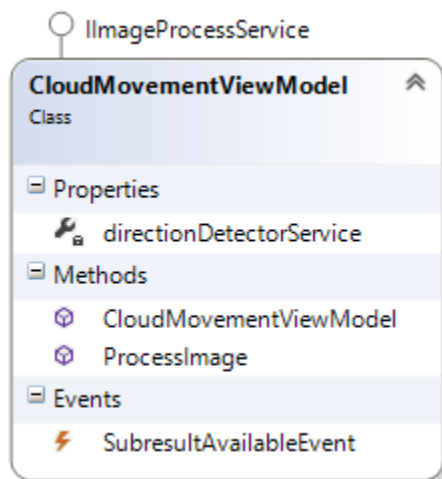
10.8. ábra – A felhők típusának valamint az égbolt borultságának meghatározását végző metódusok felépítése. A fenti sorban elhelyezkedő osztályok megvalósítják az *ISubProcessService* interfészt, míg az alul lévők esetében erre nincs lehetőség, mivel nem végeznek a képen módosítást.

10.6 MAGASSÁGI SZÉLIRÁNY DETEKTÁLÁSA MODUL

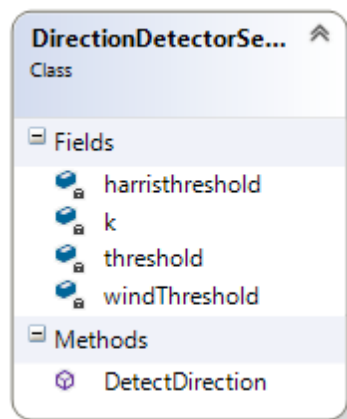
A modul lehetőséget biztosít a magassági szélirány meghatározására a képen látható felhők mozgásának iránya alapján. Egyetlen metódus biztosítja az eredményt, mely az Accord.NET könyvtár felhasználásával számítja ki az eredményt (lásd 10.9. ábra és 10.10. ábra).

Mivel a kamera nem minden esetben van a helyes irányba beállítva, tehát a kép teteje nem észak felé néz, így „Settings” menüpontban található „Wind direction” almenüpont első szövegdobozának értékének változtatásával módosíthatjuk a kamera tájolását. A megadott érték fokokban értendő, változás esetén automatikusan mentésre kerül.

Mivel nem csak a négy égtájnak megfelelően határozzuk meg a szélirányt, hanem köztes értékeket is megadunk, így a program lehetőséget ad egy érték meghatározására, ami a négy égtájtól való maximális eltérést határozza meg, amin belüli eredmények még oda tartoznak. Ezt a beállítást a „Wind direction” almenüpont második szövegdobozába írt eredménnyel állíthatjuk be. Az érték mentése itt is automatikus.



10.9. ábra – A felhők mozgásának meghatározására szolgáló alap modul szerkezete.

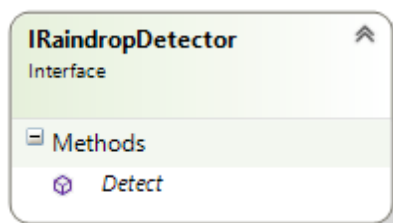


10.10. ábra – A felhők mozgását detektáló osztály felépítése. Az ISubProcessService interfész megvalósítására nincs lehetőség, mivel nem végez módosítást a képen.

10.7 ESŐZÉS DETEKTÁLÁSA MODUL

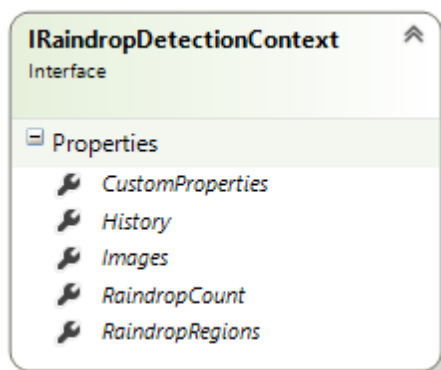
Ebben a modulban célunk képfeldolgozással lehető legpontosabban meghatározni, hogy épp esik-e az adott időpillanatban, és ebből kikövetkeztetni, hogy mikor kezdett esni, és mikor fejeződött be. Az algoritmus implementálásához elegendő volt az Aforge.NET programkönyvtár használata.

A folyamat két egyértelműen elkülöníthető fázisból áll. Az első fázis során az adott egymás után következő képeken meg kell keresnünk az esőcseppeket. Ezen műveletek elvégzésére egy olyan implementációt használunk, amely megvalósítja az IRaindropDetector interfészt (lásd 10.11. ábra). Egyetlen metódust szükséges megvalósítani, ez pedig a „Detect()”. Ez a megadott paraméterek alapján vissza fogja adni a talált esőcseppeket, ezek számát, valamint a műveletek körülményeit jellemző értékeket (naplózás, részeredmények stb.).



10.11. ábra - Az esőcseppek detektálását egy IRaindropDetector interfész implementációja végzi.

Ahhoz, hogy a cseppeket felismerjük, a megfelelő paramétereket egy interfészen keresztül – „IRaindropDetectionContext” – szeretnénk átadni (lásd 10.12. ábra). Ez az interfész felelős, továbbá a műveletek elvégzésének a során létrejövő tulajdonságok, paraméterek és képek tárolására, utaztatására. Végezetül az esőcsepp régiókat is itt fogjuk visszaadni. A cél a bővíthetőség, így bármely közbeiktatott – az eredetit kiegészítő – lépés így bármikor elérheti és módosíthatja a paramétereket.



10.12. ábra - Az esőcseppek detektálásához szükséges paramétereket egy IRaindropDetectionContext interfészen keresztül adjuk át. Ugyanezen az interfészen keresztül utaztatjuk a folyamat során keletkezett paramétereket és a végeredményt is.

Az IRaindropDetectionContext paraméterei a következők:

- CustomProperties – Ez egy IDictionary típus, melyben olyan kulcs-érték párokat tárolunk, mely a folyamatban felmerülő különböző értékeket – és természetesen a hozzájuk tartozó elnevezéseket – tárolja.
- History – Egy egyszerű karakterláncokat tartalmazó lista, mely csupán arra szolgál, hogy egy későbbi naplózást implementáló rendszer az algoritmus lépéseit perzisztálja, adott esetben szimplán megjelenítse a képernyőt, ha erre lenne szükség.
- Images – Szintén egy IDictionary típus, melynél az értékek Bitmap típusúak. Ide kerül az összes kép – beleértve a forrás és valamely végső képet. Az egyes előfeldolgozási lépések is innen érhetőek el.

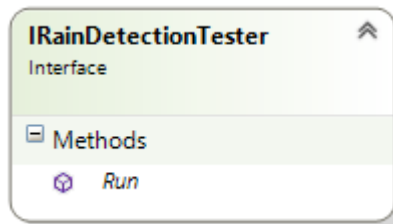
- RaindropCount – Egy egyszerű 32 bites integer típusú szám, mely a legvégső műveletek elvégzésekor kap értéket. Ez ad információt arról, hogy hány darab esőcseppet detektált a rendszer az aktuális képen.
- RaindropRegions – Végezetül egy Rectangle típusú lista, melyek a típus jellegéből fakadóan koordinátákkal jelöli, hogy hol lettek detektálva az egyes cseppek.

A kontextusban két féle kulcs-érték párt tartalmazó listánk van, egyik amely az egyedi tulajdonságokat tárolja, másik pedig a létrejött képeket. Ahhoz, hogy minden modul könnyedén megtalálja a számára megfelelő objektumot, ahhoz egy konstansokból álló kulcs listát volt célszerű bevezetni. Ez természetesen közel sem azt jelenti, hogy csak ezekre a kulcs értékekre lett leredukálva a lista, viszont az alapvető lépések végeredményeit egy előre definiált kulcs mögé tesszük. Ezen kulcsok a következő konvenció alapján kerültek elnevezésre: Gyűjtőnév.Elnevezés. Például a „SourceImages.Latest” kulcs mögé kerül, az időben legkésőbb készül esőcseppek detektálására szánt kép.

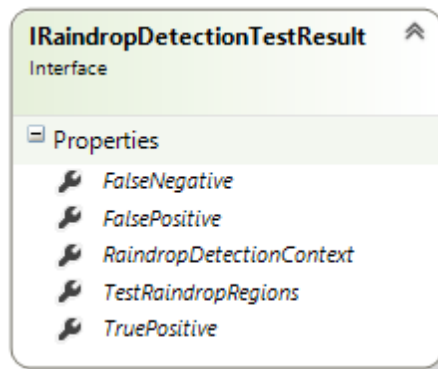
Az előfeldolgozás számos olyan képfeldolgozási műveletet tartalmaz, mely az egyes lépések bemenő paramétereinek módosításával egészen más eredményt érhetünk el. Az egyes paraméterek finomításának a teszteléséhez bevezettük annak a lehetőségét, hogy a felhasználó egy képen be tudja jelölni a cseppeket. Így a különböző értékek alapján lefutott algoritmus eredménye összevehető az elvárt képpel. Erre egy IRainDetectionTester interfész implementációja hivatott (lásd 10.13. ábra). Ennek egy egyértelmű metódusa van, a „Run()”. Ez két paramétert vár, egyik a már megismert kontextus, a másik pedig egy Rectangle lista, mely a felhasználó által bejelölt cseppek alapján jött létre.

A tesztelés végeredményét egy „IRaindropDetectionTestResult” tárolja (lásd 10.14. ábra). Paraméterei a következők:

- RaindropDetectionContext: A paraméterül adott IRaindropDetectionContext kiegészítve a tesztelés során létrejött képekkel. Ilyenkor lehetséges létrejövő kép az eredeti kép színes keretekkel a cseppek körül.
- FalsePositive: 32 bites integer, mely a nem igazi esőcseppek detektálásának a számát tartalmazza.
- FalseNegative: 32 bites integer, mely a nem detektált esőcseppek számát tartalmazza.
- TruePositive: 32 bites integer, mely a sikeresen detektált esőcseppek számát tartalmazza.
- TestRaindropRegions: A felhasználó által megadott régiók.



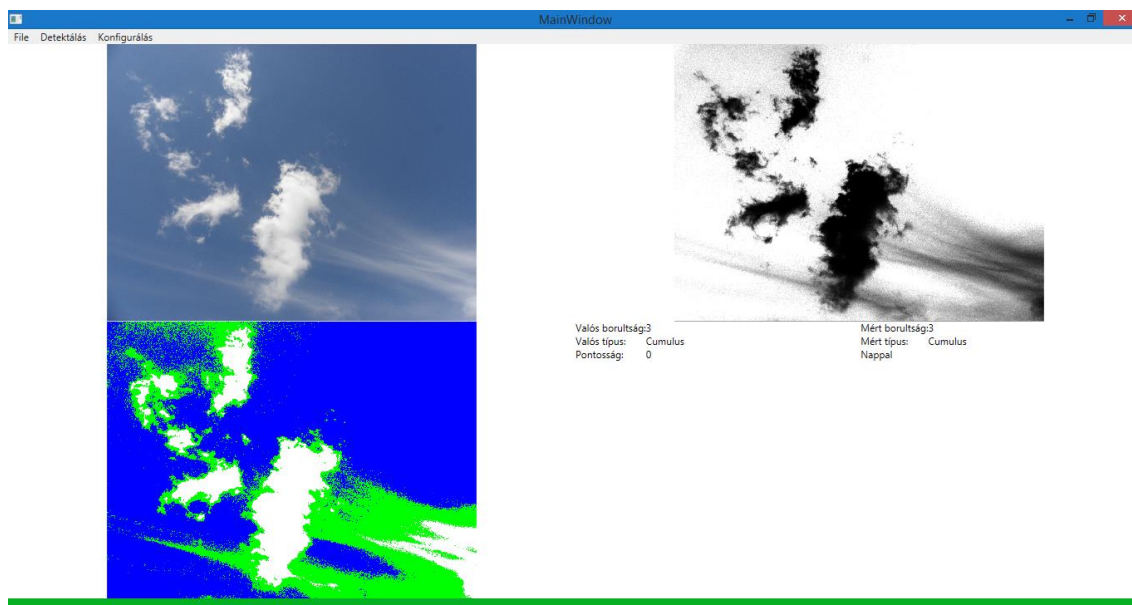
10.13. ábra - A felhasználó által tesztelés céljából bejelölt esőcseppeket a képen az *IRainDetectionTester* interfész implementációja veti össze. Így értékelhetők az egyes algoritmusok.



10.14. ábra - A felhasználó által megadott esőcseppek és az algoritmus által detektált cseppek összevetésének az eredményét egy *IRaindropDetectionTestResult* interfészen keresztül adjuk vissza.

10.8 FEJLESZTŐI VERZIÓ

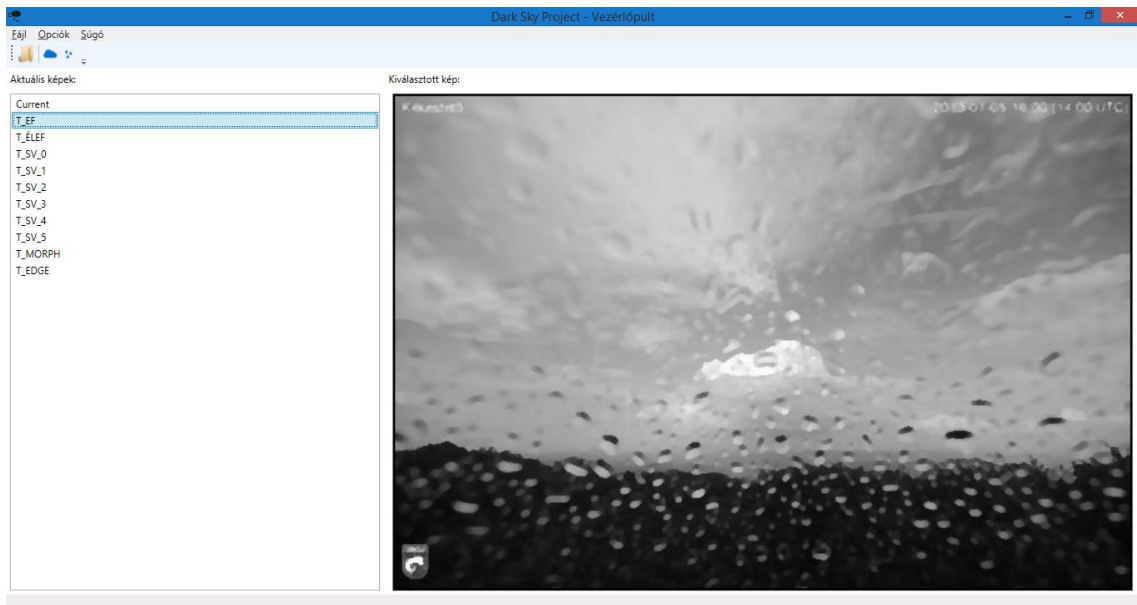
A fent leírt rendszer mellett megvalósításra került az ég borultságát illetve felhők típusát meghatározó-, és az esőcseppeket detektáló modul esetében egy külön program, ami a könnyebb átláthatóság érdekében az eredményeket és az egyes részeredményeket grafikusán is megjeleníti. Ilyen módon pontosabban követhetjük a működés egyes lépéseit, könnyebb meghatározni a hibás működés helyét, kiküszöbölni az azt előidéző tényezőt. Habár az éles verzióban erre nincs szükség, a fejlesztés során ez egy rendkívül fontos funkció.



10.15. ábra – A borultságot és felhő típusát figyelő modul felhasználói felülete grafikus megjelenítés esetén.

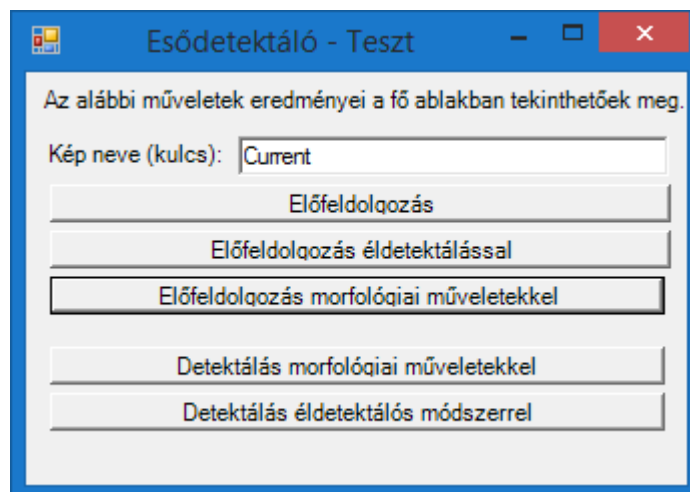
Az ég borultságát, illetve a felhők típusát meghatározó modult az „Opciók” menüpont, „Felhők detektálása” almenüpontjával nyithatjuk meg. Amennyiben már lett megnyitva kép, az automatikusan betöltődik, ha nem, akkor a „File” menüpont megfelelő parancsának kiválasztásával tehetjük meg. Lehetőségünk van egyetlen kép és sorozat megnyitására, valamint képek megnyitására statisztika készítéséhez. Az ablakban három kép jelenik meg (lásd 10.15. ábra). A baloldalon felül található az eredeti, míg a jobbra lévő a szaturációs és elmosott kép. A bal alsó részen a küszöbölt kép látható. A grafikus megjelenítés mellett még egyéb a fejlesztést, dokumentálást, tesztelést segítő funkciók is kerültek. Ilyen a mért adatok összehasonlítása az érzékelők által megadottakkal. Kijelezve az ablak jobb alsó negyedében van, így az eredményekről egyből információt kaphatunk. Ezt a rendszer képes több képen vagy akár sorozaton elvégezni, majd minden egyes eredményt egy szöveges fájlban tárolni. Ezt az értéket a „File” menüpontban található szövegdoz értékének változtatásával állíthatjuk be. Minden importált adat tartalmazza a kép nevét, a szakemberek által megadott borultságot és felhő típust, valamint a program által mértet. Sorozatképekkel való tesztelés esetén lehetőségünk van megadni, hogy a program hány képet számítson egy sorozatnak. Ezzel az első és utolsó kép készítése között eltelt időt befolyásolhatjuk, ami újabb tesztelési lehetőségeket biztosít ezen a téren. Ez a funkció a program működésének kiértékelését nagymértékben gyorsította, mivel akár ezer képen végrehajtott folyamatokról is egyszerűen készíthetünk statisztikát. Mivel ez egy hosszú folyamat, az ablak alján egy sáv lett elhelyezve, ami jelzi a feldolgozás állapotát. Azonban előfordul, hogy nem automatikusan szeretnénk teszteket végezni, hanem egyenként az általunk kiválasztott képeken. Ennek segítésére található a jobb oldalon

kijelzett információk között egy „Pontosság” érték, mely minden automatikusan futtatott feldolgozás után az eredménynek megfelelően változik.



10.16. ábra – Grafikus rendszerünk működése esőcseppek detektálása közben. Látható, hogy milyen útvonalat járt be az algoritmus, ameddig a végső érték meg nem született.

Hogy egyszerűbb legyen megfigyelni, hogy a feldolgozás mely része okoz esetleges hibát a végeredményben, a detektálást a felhasználónak lehetősége van lépésről lépésre lefuttatni. Ezt a „Detektálás” menüpont almenüpontjaival lehet megtenni. Ebben az esetben azonban fontos a helyes végrehajtási sorrend betartása a megfelelő eredmény érdekében. Ezt könnyen betarthatjuk, ha a menüpontok között fentről lefelé haladunk. Köztük az utolsó az automatikus detektálásra szolgál, amivel egy gombnyomással végezhetjük el a detektálást. Minden műveletek végrehajtásában nincs változás a szöveges felülethez képest. A funkciónak köszönhetően volt lehetőség különböző beállításokkal való tesztelésre is, ami alapján fontos következtetéseket vonhattunk le a fejlesztés során.



10.17. ábra – Funkciók végrehajtása egyenként grafikus esődetektálás esetén.

Az esőcseppek detektálásáért felelős modul az eredeti ablakban fut, ami két részre van osztva. Baloldalon egy lista látható, ami a feldolgozás egyes lépéseit tartalmazza, míg a jobb oldalon mindig az aktuálisan kiválasztott listaelemhez tartozó kép van megjelenítve (lásd 10.16. ábra). Első lépésként egy képet kell megnyitnunk, amin a detektálást végre szeretnénk hajtani. Ezt a „File” menüpont „Megnyitás” parancsával tehetjük meg. Kép megnyitása után az „Esőcseppek detektálása” gombra kattintva kezdhethetjük meg a műveleteket. Azonban ez teljesen manuálisan működik, egy újabb felugró ablakban haladhatunk végig a lépéseken (lásd 10.17. ábra). Ha egy művelet kiválasztása után szeretnénk megtekinteni a végeredményt, a főablakra kattintva a megfelelő listaelem kiválasztásával a műveletvégző ablak bezárása nélkül megtehetjük.