

MATCHMOVE

PFTRACK-EL ÉS MAYA-VAL



+

PF Track™ 5.0

A New Dimension in Tracking

+



2



HA EZT OLVASOD MATCHMOVE -OLNI FOGSZ.

Ezt a dokosit a pfTrackre 4.1 release 3-ra, és azokra a lépésekre próbálom kihegyezni amik nincsenek benne a hivatalos doksiban, vagy nem elég részletes a leírásuk. A pfTrack hivatalos doksija itt:

quickStart guide: http://thepixelfarm.co.uk/Support/Manuals/PFTrack_4.1_QS.pdf

step by step tutor: http://thepixelfarm.co.uk/Support/Tutorials/Track_Match/GettingStarted.pdf

documentation: http://thepixelfarm.co.uk/Support/Manuals/PFTrack_4.1_Ref.pdf

további ajánlott irodalom:

http://www.amazon.co.uk/Matchmoving-Invisible-Art-Camera-Tracking/dp/0782144039/ref=sr_1_1?ie=UTF8&s=books&qid=1232553239&sr=8-1



MI A MATCHMOVE?

A valóságban a jelenetet rögzítő kamera összes, a kepalkotas szempontjából fontos tulajdonságot keresed azzal a céllal, hogy a forgatott footage-hez 2D-s vagy 3D-s elemeket ugy lehessen illeszteni, hogy a néző számára az egész terben és mozgásban egy időben, egy kamerával felvett képkent hasson.

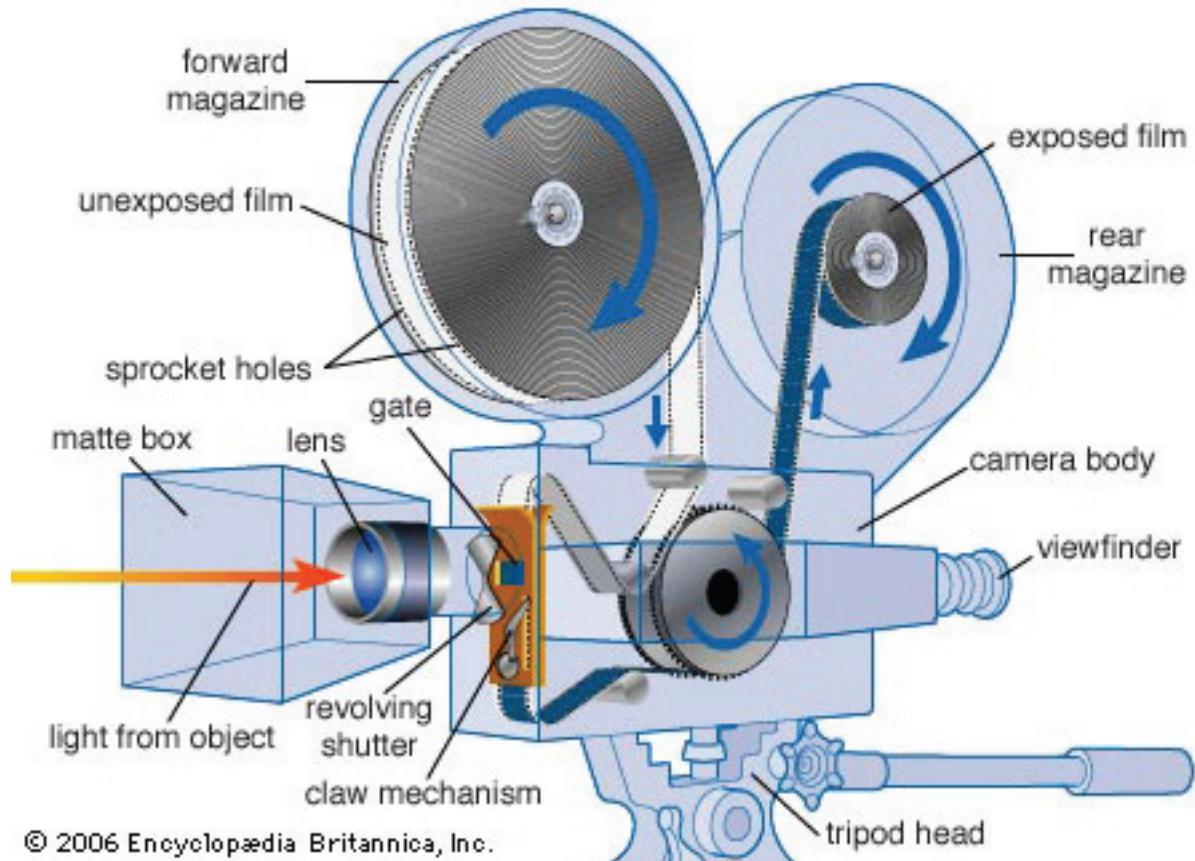
EZEK A KAMERA TULAJDONSAGOK

ERDEKELENEK - MINDEGYIK:

- focal length,
- film back / camera aperture
- mozgas
- forgas

A **focal length** es a **film back** specialis abból a szempontból, hogy a matchMove számolásnál – erről kesőbb – minden adatot ki lehet számolni – illetve a film backre a szoftver használni fog egy alapértelmezett értéket – viszont ezek elterhetnek a valós adatoktól ugy is, hogy jo a matchMove, de a perspektiva mozgás közben még hogy változik majd. Ezt a ket adatot ezért a matchMover meg szokta kapni. A film back-et általában jól, az egy filmnel allando konnyen megjegyezheto parameter, minden shoton ugyanaz, de a focal length-et forgatáskor shotrol shotra, takerol takre, stb... fel kene valakinek irnia es ide be-becsuszsnak elírasok.

A cel a forgatásnak használt kamera lehető legfontosabb digitális megfelelőjét megtalálni digitális



© 2006 Encyclopædia Britannica, Inc.

kepi elem letrehozasahoz es a jelenethez passzintasahoz uly, hogy a valós és a digitális, technikailag, fizikailag máshogy létrejött képi elemek között ne legyen észlelhető különbösz perspektívában, térbeli elhelyezkedésben es mozgásban.

Felhasznalasa leggyakrabban 3D -s rendernel tortenik, de job Jon kamera projection-hoz, illetve kompozitalashoz is, pl.: segithet cornerpin-es cleanPlate illesztesnel ott, ahol a footagen nincs a cornerpin-hez kello szamu track, vagy ha van is, nem latszanak eleg hosszan vagy szunet van ott. Ezeket leirom kesobb.

MATCHMOVE A GYAKORLATBAN

MATCHMOVE WORKFLOW/PIPELINE:

filmScan --> undistort --> camera adatok, survey adatok bele a szoftverbe --> 2D/3D track --> 2D track ellenörzés / javítás --> solve --> javítás amíg szükséges és lehetséges --> scene letrehozása 3Dben - kamera a megfelelő helyre a megfelelő méretben --> 3D department

A matchMove – mint munkafolyamat – lenyege, hogy a matchMove szoftvernek a lehető legtöbb a jelenetben levo **valós** – erről kesobb –, **nem mozgo terbeli pontot**



megmutasd a leheto legszelesebb skalan a leheto legtobb frame-en. Ez a trackeles – kesobb reszletezem. A legszelesebb skala alatt a footage es a jelenetben levo ter leheto legjobb lefedettsege ertendo. pl.: hiaba van 100 jo track-ed a bal felső sarokban, ha a jobb alsot kell kiegészíteni, illetve ha tavolra, az utca vegere kell elemet betenni, nehezebb lesz, ha csak a kozel levo hazakat trackelted.

Ha jol csinalod a szoftver vissza fogja tudni szamolni a kamera paramtereit, a mozgasat es a teret – koordinata rendszert – sot, a trackek jo reszet el is fogja tudni helyezni a kamerahoz kepest, ami azt jelenti, hogy tudni fogod, hogy a terben hol lesz pl. a haz sarka, az ablakok, a festes az utoron, alvo fekete macska, stb..., ami segitseg abban, hogy ami elemet be akarsz illeszteni a megfelelo meretben a megfelelo helyre tudd tenni.

Trakelesnel az a celod, hogy **minden egyes lerakott track az osszes frame-en ahol latszik ugyanarra a terbeli pontra mutasson mint amikor leraktad.** Ezek lesznek azok a pontok amik alapjan a celszoftver, – pl.: boujou, pfTrack – a photogrammetria – <http://www.photogrammetry.com>, <http://en.wikipedia.org/wiki/Photogrammetry> – matekja alapjan dolgozni fog. Ez a resz a solve, mint megoldas a trackekre a photogrammetriai egyenletben – errol jellemzoen nem akarsz tudni lehetoleg semmit, es nem is nagyon kell –, de neha az eredmenyt is igy szolitjuk.

Valos – fix, a teljes shot-on ugyanabban a terben ugyanazon a helyen marado – pontokat megmutatni nem mindig egyertelmu, fontos, hogy a bizonytalan trackeket ellenorizd – errol kesobb – es ha rossznak bizonyulnak kidobd! A bizonyosan rosszakat termesztesen azonnal – erre leirok kesobb a pontonkenti guberalasnal jobb moszert! A kovetkezo pontok szamitanak kockazati tenyezoknek:

ARNYEKOK SARKAI:

nem egyertelmu rossz trackek, **hasznalhatosagukat az hatarozza meg, hogy a jelenet alatt elmozdul-e az arnyek.** Pl.: ha egy haz arnyekot vet az utcara, es a fenyforras amit a haz kitakar a nap, nagy biztonsaggal ki lehet jelenteni, hogy a nap akar 300 frame-en sem tekintheto mozgo testnek ezert uyanoda veti az arnyekot – ha a haz mozog valamelyen oknal fogva akkor persze fakabat. Altalanossagban az arnyekok sarkai hasznalhato pontoknak szamitanak mindaddig, amig a fenyforras, az arnyekot vovo test es amire az arnyek vetul vegig ugyanazon a helyen marad, sem egymashoz, sem a keresett terhez / koordinata rendszerhez viszonyitva nem mozdulnak el.

HIGHLIGHTOK:

mivel a csucsfenyek fuggenek a feny beesesi szogetol, **soha nem mutatnak frame-eken keresztul valos terbeli pontra.** Csak abban az esetben lehet hasznalni oket, ha a kamera csak forog vagy a targy amirol vetodnek olyan kis teruletet fed le a kepen, hogy az elmozdulas hibahataron belul marad. Pl.: 10-15 pixeles auto, 4-6 pixeles ablakai a hatterben. Ha csucsfenyeken kivul van elegendo pont a solve-hoz biztonsagosabb kerulni oket.

FELULETEK, ELEK TALALKOZASAI:

szinte sosem jo pontok. pl.: ket haz egymast mogott kameramozgas kozben mas utat jar be a photon. A tavoli lasabb mint a kozolebbi. Ha van eluk ami talalkozik altalaban jo trackelheto pontot ad, de minek, a kulombozo sebessegu mozgas miatt a trackelt metszespont mas terbeli helyen lesz kockarol kockara.

FORGASTESTEK:

legtobbszor nem jo pontok. pl.: egy henger teteje oldalrol nezve saroknak latszik, trackelni is jo lehet, de ugyanott lesz a kepen a pont megha koruljarja is a kamera a hengert,

tehat nem valos terbeli pontot szokott mutatni. pl.: viztorony oldalrol – a tetejen valoszinuleg lesz egy jo trackelheto pont, de ha a kamera koruljarja ugyanott lesz a track vegig a shoton, mikozben a kamera mozgott es fordult.

Gombok abbol a szempontbol specialisak, hogy ha eleg kicsik a shoton az egeszet egy nagy trackpontkent felfogva jo hasznalhatoak lehetnek. pl.: teniszlaszti nem tul kozel kifejezetten segithet trackelni. Legalabbis ha betartja az elso szabalyt es fixen never valahol a shoton es nem eppen jatszanak vele.

A JO TRACKEK OSSZESSEGE SOLVE SZEMPONTJABOL:

Hiaba van sok jo tracked ittis ott is, hosszabban rovidebben, ha nem figyelsz a lepcsozessegre – lejjeb kifejtem – azzal nagyban megnehezited a pontos solve-ot.

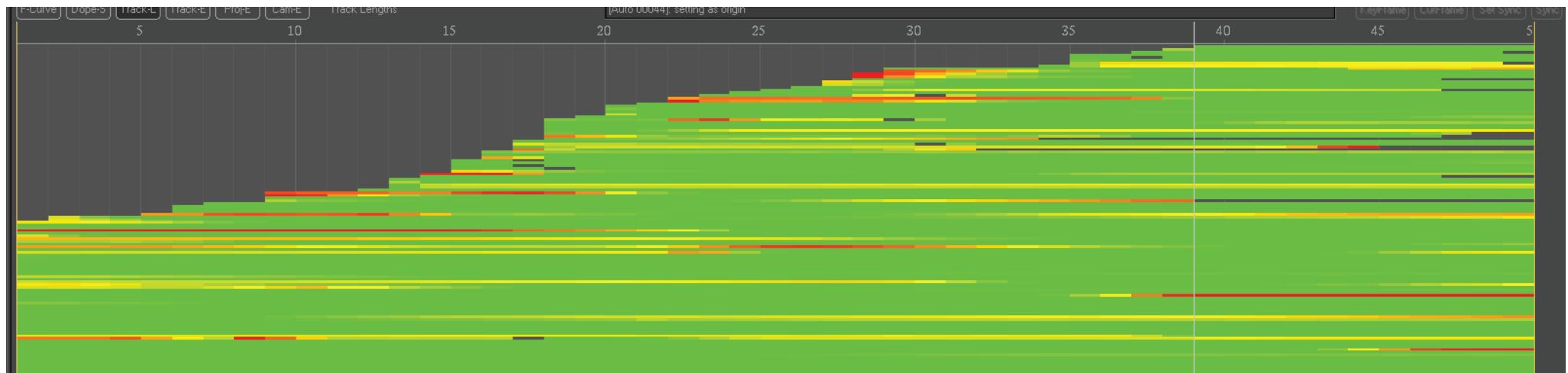
Ha a trackeles reszt letudtad, a solve ugy mukodik, hogy **a szoftver keres 3 frame-et, ahol a legegyszerubbent tud olyan kamerat kiszamolni, hogy az adott frame-eken a leheto legtobb pontot a leheto legpontosabban egy terbe tudja illszteni**. Ezzel megkapja a teret, a kemara helyet, es a pontok jo reszenek a helyet a teren belul. Ezutan elkezdi kiszamolni a kamerat, kulon kulon framenkent, majd osszekoti a kamerakat egy mozgo kamerava ugy, hogy kozben finomitja mozgast aszerint, hogy az uj framen ahol tart a pontok ugyanott legyenek ahol az elozo es a kovetkezo fram-en. Ez az „**initial solve**”. A dolog termeszetebol fakadoan – nincs tokeletes track pl. a zaj miatt, feltunnekk pontok, stb... – a solve mindig ezen a 3 frame-en belul a legpontosabb. Ha ez megvan folytatja framenkent ugy, hogy minden további framet az initial solve-hoz igazit.

Kezenfekvonek tunik belepiszkalni az initial solve-ba - pfTrackben bele lehet – ugy, hogy az elso, a kozepso es az utolso framekkel kezdje, hogy vegig pontos legyen es orulunk, de ennek vannak korlatai. Az initial solve ugy lett kitalalva, hogy azok kozott a bizonyos kezdo fram-ek kozott legalabb 6 pontnak vegig latszania kell. Ha tobb megy vegig az jobb, de 6 a minimum. Ezzel biztositva van, hogy a solve ezeken a frameken a leheto legpontosabb teret adj es a pontok is a legjobban a helyukre keruljenek. Es itt jon kepbe a lepcsozesseg.

Mivel **a matchMove alapja, hogy mozgas van**, es a pontok elmozdulasabol szamolja vissza a dolgokat, eleg valoszinu, hogy lesz pontod ami csak a shot elejen latszik, lesz ami csak a vegen, lesz ami csak a kozepen, es lesz ami ugyan vegig menne, de itt ott betakarnak dolgok hosszbb rovidebb ideig. Ez teljesen termeszes lesz. Amire itt erdemess figyelni, hogy **ha egy pont eltunik, legyen helyette masik**, ami info-t ad a solvernek az eltunt pont koruli terrol.

Irtam, hogy kell az initial solve-hoz 6 vegig lathato pont. Ez a shot teljes hosszara igaz, azzal a kulombseggel, hogy nem kell majd mindig minden pontnak vegigmennie, de ahoz, hogy egy 2D-s kepbol teret lehessen szamolni 6 pontnak minimum kell lennie az egyenletben. Az initial solve kozben es utan tele leszel hosszabb rovidebb pontokkal. Mivel minden egyes frame-en lehet keresni egy kamerat, kamera parametereket es teret ami jo solve-ot ad, fontos, hogy ezek kozott idoben is legyen osszefugges, hogy amikor a solve osszekoti a fame-ekre kiszamolt kamerakat ne legyenek ugralasok, tores, stb. Ezt az osszefuggest a framek kozott tudod biztositani a lepcsozesseggel, ami szimplan annyit jelent, hogy odafigyelsz ra, hogy a trackek a legheto legtobb fram-en, szepen egymasba simulva irjak le a teret a solvernek.

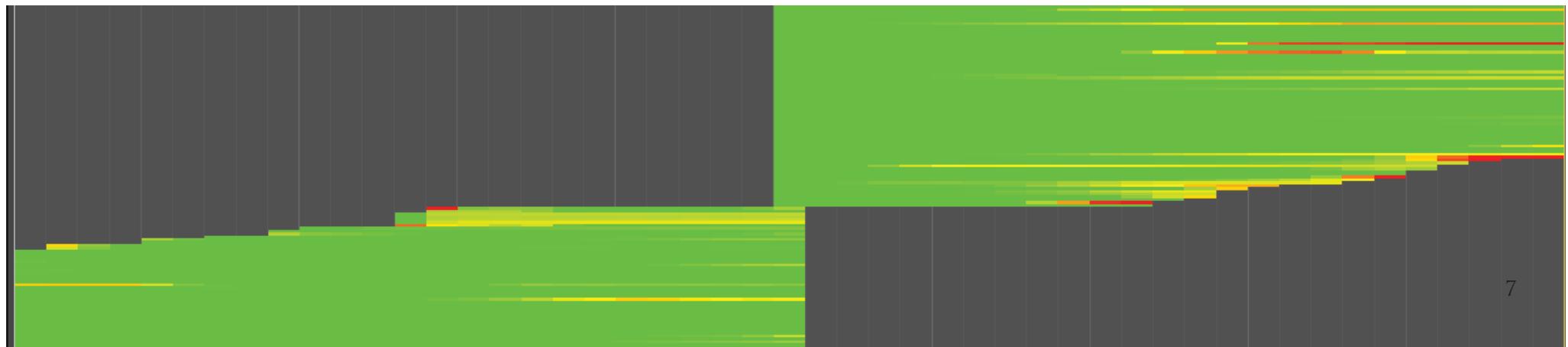
A pfTrackben ezt a „track length” fulon, a footage alatt nagyon szepen lehet kovetni. Erdemes hosszusaguk szerint sorba rendezni a trackeket az alapertelmezett sorrend helyett („,0” gomb a jobb szelen.)



LEPCSOZETESSEG FONTOSAGA:

Van 20 tracked vegig. Van meg 10 ami csak a footage kb feleig megy, van meg 10 ami kb a footage felettel. Ha nem talalsz ki valamit amivel ezt a törést lekezeled valoszinuleg ugrani fog a matchMove, mert szamolas kozepen a teret leiro pontok 30 %-a kiesik, hogy bejonn ujabb 30% ami mashogy irja le a teret. Ilyenek ellen legvegso esetben meg azt is erdemess bevetni, hogy a nem vegigmeno 10-10 tracket siman kidobod, es csak az eredeti 20-al dolgoztatod a geopol. Ez boven lehet hogy ugyanolyan pontos lesz, viszont ugras nelkul. **A legjobb megoldas persze, a lepcsozetesseget elerni, vagyis ameddig lehet kezzel meghosszabbitani a trackeket, hogy fedesbe keruljenek legalabb 2-3 frame-en.** Siman lehet hogy a fordítottja fog segiteni, es ha nagy nehezen parat fedesbe hoztal, a tobbivel nem kuszkodsz, hanem szepen elkezded lepcsozetesen kidobni oket minden irányba, hogy egyenletesen dolgozhasson a solver. Az aranyok persze fontosak, nem kell tulzasba vinni, ha van 150 tracked a footagegen, es a jol trackelheto peepshow neon balra el, 10 pont ugrott, a maradek 140 eleg nagy biztonsaggal ad jo teret a solvernek es nem lesz ugras.

A LENCSE TORZITAS (DISTORT/UNDISTORT)



Mielott vadul nekiallnal trekkelní, egy dolgozatban meg amit matchMove előtt meg kell oldani – ha lehet. **Minden lencse torzít** valamennyit, ami annyit jelent, hogy a kép szelei fele, egyenletesen eltolódik a kép, néha szabad szemmel nem is lathatóan. Ez lencse hiba – inkább tulajdonság –, minden jelen van, gyártorol gyártora, lencserol lencsere, sorozatszámrol sorozatszámra, sőt focal length-ról focal lengthre változik. A halszem optikához hasonló, de normal lencseknel, messze nem akkora a torzítás, de arra azért megis elég szokott lenni, hogy az az 5-10-20 pixeles elmozdulás a kép szelei fele megveszesse a trekkjeid, ami elég sok ahoz kepest, hogy mint azt majd a solve-nal kifejtem 0.6 pixel alá szeretné pontosítani a legnagyobb hibát.

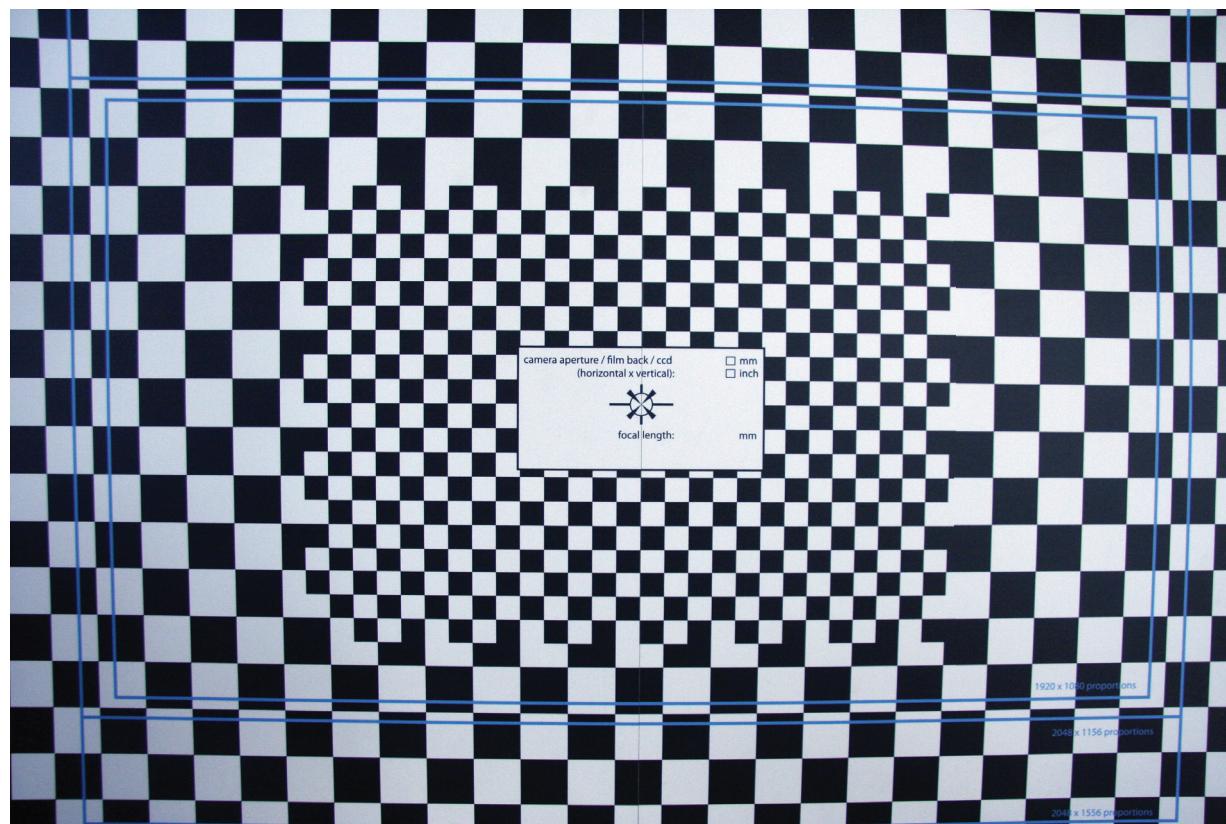
Lencstorzítás allen egy dolgot tehetsz. minden forgatáson fel szoktak venni egy negyzetracsot – lens grid – minden lencséhez amit használni fognak, minden focal length-el. Ebből vissza tudod számolni mekkora a **torzítás amit ezután ki kell szedned a footag-ból** amit trekkelsz.

A matchMove szoftverek erre általában adnak megoldást, de **ajanlott abban a kompozitor szoftverben megoldani ezt, amiben aztan a kompozitorok, dolgozni fognak.** Oket nem zavarja majd a lencstorzítás, és nem is kitorzítgatott anyagon, hanem az eredeti dolgozniak, viszont, hogy passzoljon a CG, vagy a clean plate, arra vissza kell tudni rakni a lencstorzítást, hogy passzoljon a footage-hez. Ha kompozitor szoftverrel undistortoltal, akkor ugyanazt a node-ot fogják használni, csak atkapcsolják undistort-rol distort-ra és berakják a CG agba. Ha más szoftvert használ, hozzázz a teljes CG-t, cleanPlate-t külön le kell renderelni csak a torzítás miatt, ami macera és helyet foglal.

Ha veletlenül nem kapnal lens gridet, matchMove-olj analkeul. A footagek egy részénél esetleg meg lehet próbálni kiszéjni a torzítást, de ha nem csupán egyenes vonal van a footagen, pl.: new yorki felhőkarcolók, hanem szabálytalan felületek, pl.: erdei, direkt girbegurba agakkal egy sziklalomás szélen eselyed sincs. Szemre ne probálkozz, akár hogyan nezed sosem lesz jó, és sosem fogod tudni eldönteni, hogy a trekkek és a solve-od rossz, vagy a lens distortion vezette-e meg. - ugyis a lensdistortion-re fogod fogni (joggal).



Torzított footaget általában siman le fogsz tudni trekkelní, ha van eleg pont kicsit maszkold le a kép szelet, hogy ott ne legyen trekk ahol már jobban torzul. A baj csak az lesz, hogy ilyenkor a solve - erről később - kamera mozgasnak fogja latni a mozgást a trekkekben. Ez ellenőrzéskor nem szokott kiderülni, 3D-ben sem minden, de amikor összerakja a kompozitor elcsuszik majd a footage széle fele a dolog. Ha ilyen van matchMove-olva le, kompozitban muszály lesz kitalálni valami kis lencsetorzítás.



TRACKELES

valoszinuleg nem eloszor hallasz trackelesrol, a folyamat ezen resze ugyanaz lesz mint mashol (2D).

MatchMovenal hasznalni fogsz egy advancedebb technikat is (autoTrack), de errol kesobb. A dolog lenyege, hogy kinezel egy jo kis pontot a shoton. es lekovetteted a mozgasat kockarol kockara a szoftverrel. A korábbiak alapjan a jo kis pont olyasvalami, ami nem mozdul el a kordinata rendszerben, es a lehető legtobb kockan latszik.

Ez az egész ugy nez ki a gyakorlatban, hogy kivalasztod a user feature toolt, raboksz egy frame-en a jo kis pontra es raszolsz a pfTrackre hogy keresse meg ugyanazt minden frame-en. Sajnos sokszor nem fogja. Ha ilyen van vagy megall vagy nem all meg, csak nem azt talalja meg amit eloszor megmutattal neki. Ha ilyen van keresd meg az utolso frame-et ahol jo es probald ujra elinditani. Ezzel uj pattern adtal neki es lehet azzal mar szepen tovabb fut. Erdemes a track parametereket is finomhangolni. Ha ezutan sem megy vagy uj szakmat valasztasz vagy tovabblepesz.

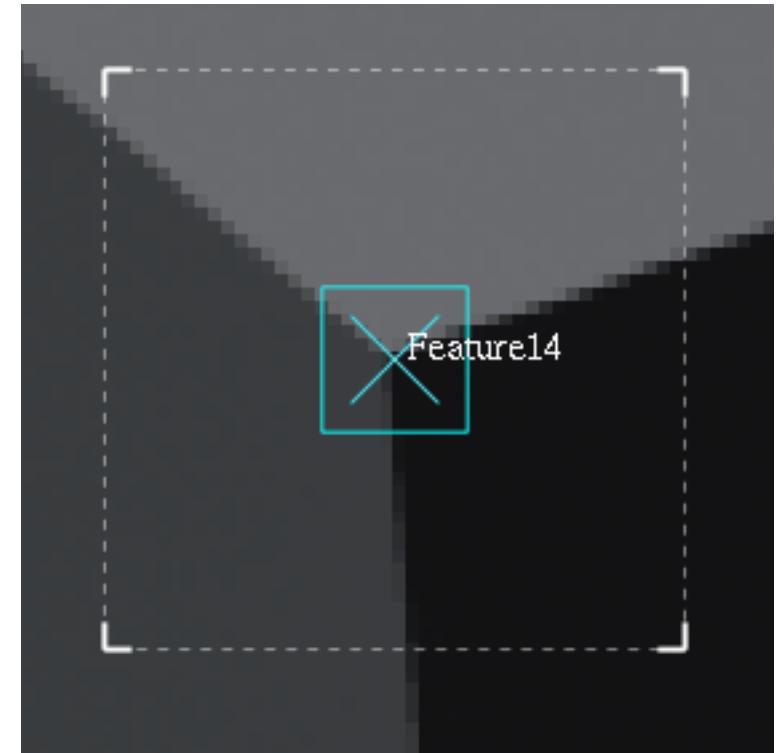
Tovabblepni ugy tudsz, hogy ha van eleg trackelheto pont siman keresel egy ujat, illetve ha nincs, kezzel kiegészíted framenkent amit a pfTrack nem talalt meg meg. Ez persze csak akkor lehet, ha a szoftver nem talalja a mintat, vagy sok a hasonlo egymas mellet es mindig atugrik, de te egyertelmuen meg tudod mutatni neki, hogy melyik volt az. Ha a track kimegy a kepbol, tul bizonytalan megmutatni, vagy eltakarja valami nyugodtan hagyd ugy.

Ha ilyen elakadas van, **2D-ben lehet a trackeket offSetelni, vagyis a patternt athelyezni egy kozeli masik pontra ami tovabb latszik. Mivel a matchMove a valos terbeli fix pontok mozgasara a shoton van epítve, ennek nincs ertelme matchMove-nal.** Egyik matchMove szoftverben sem lehet offzetelni. Vagy megmutat egy track egy pontot a terben vagy nincs ertelme a tracknek.

A trackelest valamennyire **tudod parameterezni, es ez mindig mindenhol nagyon fontos lesz.** Az advanced technikanal ugyanugy. Amennyire tapasztaltam oktatas alatt es melonal, a rossz, nehezen kezelheto solve-ok jo reszenel siman meg lehetett volna uszni egy csomo troubleshootot, maya bindzsizest ha az illeto finomhangolja a trekkelest es nem csak elengedi az autotrack alapertelmezett parameterekkel, hogy majd csak lesz valami 500 pontbol!

AUTOTRACK:

Az autotrack az amikor **a matchMove szoftver megprobal helyettesi letrekkeli pontokat**, hogy ne egyesevvel, kezzel kelljen veszodnod. Ugy mukodik, hogy minden fram-en megkeresi azokat a pontokat amelyek megfelelnek a parametereknek – ezekrol kesobb – amiket beallitottal. Ha ez megvan szinten a parametereknek megfeleloen elkezd osszefuggeseket 10 keresni a fram-eken levo pontok kozott, es megprobalja osszekotni őket egy hosszabb osszefuggo trackhez.



Ez rengetek munkat megsporol neked, de generál is ugyanannyit. Mindenesetre a hibaival együtt szeretni fogod, és azokat a shotokat kiveve, ahol előre ránézesre is vadaszni kell a pontokat ezzel fogsz kezdeni mert mindenkor egyszerűbb lesz kikukazni a rossz pontokat mint kezzel letrekkelní egyesével 100 pontot.

TRACK PARAMETEREK

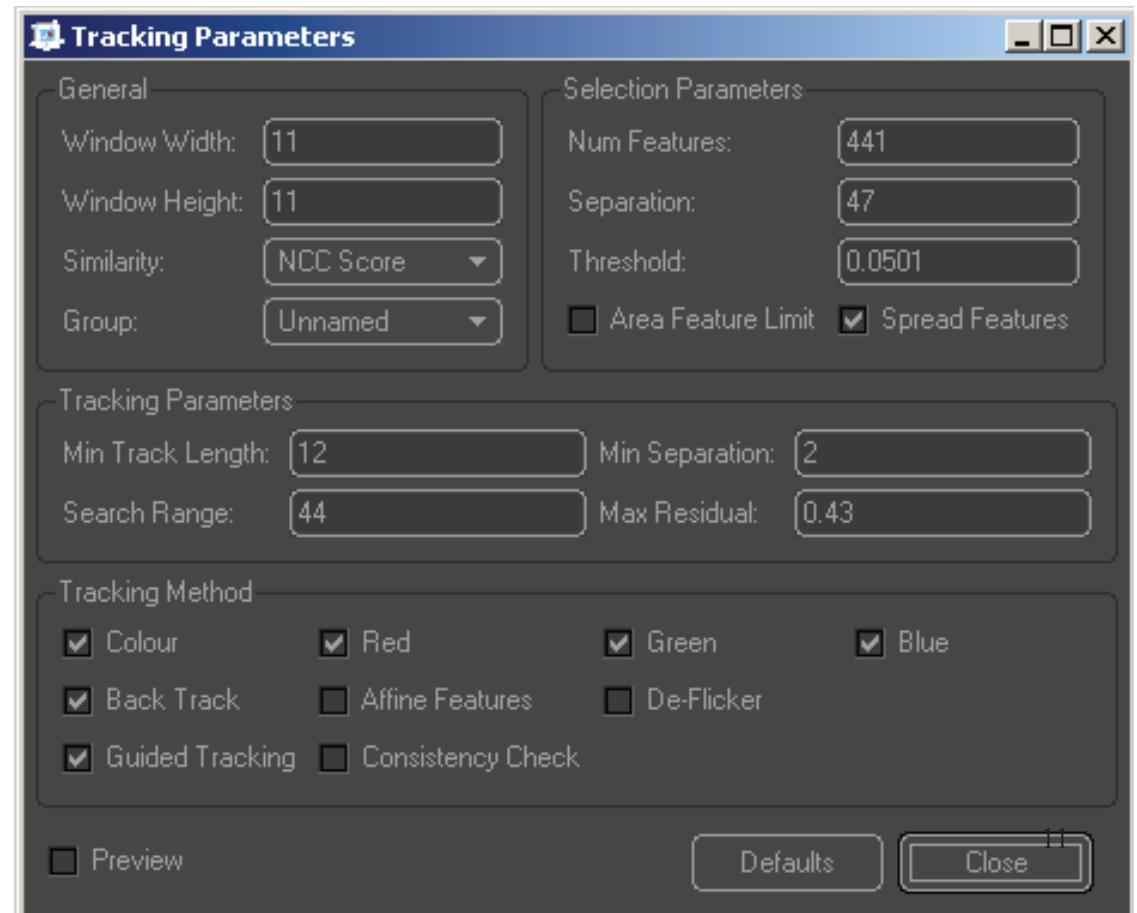
Egy track-nek 3 része van. Van egy középpontja, ami a jó kis pont maga. Ez korül van egy vékony keret a „pattern area”. Ez az a terület amit a szoftver kockával kockára keresni fog, lehetőleg a ponttal a közepen. A szaggatott keret ezeken kívül a „search area”. Ez az a terület amin belül a pfTrack a következő frame-en keresgeli fogja a pattern area által lefedett területet.

Ezeket fogod változtatni és azt, hogy milyen szempontok szerint és mennyire szigorúan ítélni a pfTrack a következő frame-en találatot.

window width: ez a pattern area. Általában a lehető legkisebbre eredményt ad leg pontosabb találatot. A legjobb, ha ugy fedi le a területet amit trekkelhetőnek ítélsz, hogy egy kicsit kilog. Inkább nagyobb legyen mint kisebb, autotracknel az átlagos pontakra lodd be. A footage-en változni fog a minta mérete alakja mozgas közben - pl. közélebb jön a elmosódik, stb... - kezi tracknel simán lehet állítgatni ahogy haladsz a footage-en. Autotracknel menet közben csak korlátosan, ezért is eredménytelen beloni az elején ezeket a paramétereiket! Ezzel a parameterrel lehet a leg jobban befolyásolni, hogy megtalálja-e amit szeretné és hogy mennyire pontosan.

similarity: ezzel a módot választhatod ki ami alapján összehasonlíta a patternt az adott frame-en azzal, amit mutattal az elején. (Semmi kulmban nem tapasztaltam meg a 2 lehetőséget között)

min track length: ez nem befolyásolja a minőséget, de sokat fog segíteni. Ezzel tudsz az auto-trackereknek egy minimum hosszúságot megadni frame számban, amivel segíted a lepcsozetes séget úgy, hogy pl. kidobatod



automatikusan a trackeket amik kevesebb mint x frame-en latszanak. pl. 2 fram-en. A tipikus ertek 7 es mondjuk 13 kozott valt be nekem. 7 frame ala csak a ketsegbeeses kerget olyan shotokon ahol nagyon nehez egyaltalan tracket talalni, nemhogy hosszut. esetleg van sok pont csak az erdoben rohannak a kameraval es meg-megszunneek pontok, de utolag ossze lehet mergelni - errol kesobb - oket egy hosszabb tracknek. 13 frame fole azert nem erdemess menni, mert ha van olyan hosszu ugyis megtartja, ha pedig van egy ket pont ami csak 7-10 framen latszik meg siman lehet jo lepcsozes a track es akkor nem art ha megvannak.

search range: ez a search area merete. ha ennel tobbet mozog a pont a footagen nem lesz meg. Ezzel nem sok jatek lesz, nagyjabol megsaccoja az ember atlagban mennyit mozognak majd a trackek es azt beirja. Olyan lesz, hogy ha tul nagy konnyebben esik hasonlo pattern a searh area-ba es konnyebben atugrik a track, de ha meg nagyon leveszed semmit nem talal majd. Ha tul nagy meg annyi hatranya is van, hogy nagyon lelassitja a trekkelest. persze ez lesz a legkisebb bajod, ha soka a mozgas fel kell venni.
num features: ezzel sincs sok jatek. Max ennyi pontot tart meg framenkent. Arra lehet hasznalni, hogy ha van 20-30 nagyon jo pont akkor azokhoz 30-ra beallitva egy szigorú tracklessel - lasd treshold parameter - eleg jol be lehet loni azt a 20-30 legjobb pontot.

separation jo kis parameter, 2 track pont kozotti minimum tavolsagot lehet megadni. Arra valo, hogyha engedunk 100 tracket mert szemre siman van annyi kellemesen magtalalhato jo pont. de van a footageen egy becsillanos, vagy barmi miatt kiemelkedoen jol trackelheto resz ahova osszegyujtene a 100 jo tracket ezzel szet lehet szoratni a pontokat a footage feluleten. Megtalalja a jo reszt is, de ezzel szolni lehet a pfTracknek hogy x pixelen belul 1 pont mar neked eleg, nezzen korul mashol is, ott is lesz jo pont.

treshold az autotrack-et lehet vele szigoritani-konnyiteni. Ez az a holtjatek ami felett trackelheto pontnak itel barmit is egy kepen. Nincs kifejezett jo ertek, shotonkent mas. Segit beloni, hogy miközben a track parameters ablak fent van, ha bekapcsolod a previw-t megmutatja mit itel jo pontnak - ez ekkor meg csak becsles, nem biztos hogy ezek lesznek meg - ha ekozben allitgatod ezt a parametert latni fogod miket talal. - neha erdemess frame-et valtani, nem mindig frissit magatol, illetve mas reszeken mas beaalitasokra lehet szukseg. Track utani rossz pont guberalas helyett erdemess ezt is jol beloni

Max residual ezzel a mar megtalalt trackeket lehet ugy limitalni, hogy ha osszehasonlitgataskor az elozo frammel e fole az ertek fole megy a track hibaja - akarmilyen mertekegyseget es osszehasonlito algoritmust is jelentsen ez - a pfTrack kidobja azokon a reszeken. A kulcsszo a kidobja, ugyanist azt nem akarod rabizni, ezert ezzel nm is erdemess jatszani kulonosebben, inkabb nagyobb legyen. Ezt leirom majd a trackles ellenorzes folyamatnal.**tracking method:**

AMIT ITT ERDEMES KIVALASZTANI:

color: a kekben van a legtobb zaj azt szinte minden alapbol kikapcsolom - kiveve ha van valami a kekben ami direkt ott latszik jol. It egy komopzitor szoftverben nezd meg melyik csatornaban lehet a legott info es azt trekkeld. - lehet pfTrackben is, de az minden cache-el es eleg lassan mutatja a szincsatornakat.

back track: annyira nem jo mint sokan vartak, altalaban nem hasznalom. Olyankor johet szoba, ha sokat mozog a kamera, jonnek mennek ki-be a pontok es elore trekkelesnel nem talal meg minden egybol csak modjuk 10 frame-el miutan feltunt vagy egyaltalan nem. Ilyenkor van, hogy hatulrol megtalalja. Ami miatt veszelyes

hasznalni, hogy visszafele trekkelesnel ugyanazokra a pontokra is rafoghat ugy, hogy mivel mashonnan kezde, mas lesz a kiindulo pattern amit keres, es ezert kicsit mas hibak kicsit mashol lesznek ugyanarra a terbeli pontra. A lejobb pelda, ha nem egyenletesen mozog a kamera, nem lesz egyenletes a motion blur. Ha felgyorsul a mozgas elnyulik a motion blur. Ilyenkor hatulrol trekkelve a motion bluros resznel tipikusan a motion blur masik veget talalja meg mint elolrol trekkelve, vagyis szetcsusznak a trekkek egy terbeli pontrol, ami ily ket kulombozo hamis mozgast fog mutatni solve-nal. Erre lehet megoldas, ha a 2 pontot mergedel - errol kesobb - de altalaban utolsok kozt veszem elo mint lehetoseget.

affine feature: hmm. elvileg finom mozgasokat engedelyez tracknel. Pl.: feher alapon fekete negyzet sarka mozog. minden frame-en uhgyanugy nez ki, csak mashol van. Ha forog is kozben, mar nem ugyanugy nez ki - mar ami a matekot illeti ami az osszhaszonlitast vegzi. Ezeket pl shake-ben lehet parameterezni, hiszen egy szigorubb feltetellel pontosabb es gyorsabb is a trekkeles. Viszont itt megint csak a 3D trekk sajatossaga, hogy szinte biztos, hogy ossze-vissza fognak forogni, mozogni torzulni a trekkek, ezert nem nagyon van ertelme nem engedelyezni ezeket a mozgasokat. Emiatt alapbol is megtalalja a pfTrack az elfordulo negyzet sarkat, igazabol nem talaltam olyan shotot ahol egyertelmuen kiderult volna ennek a paameternek a hatasa. Alapbol kikapcsolva kezdek.

De-flicker: lasd affine. Ha mas nem megy probalkozok vele, des sosem derult ki egyertelmuen mikor van ertelme, elvileg fenyviszon valtozasokat vesz figyelembe osszehasonlitaskor, ami ugyertelmezheto a doksi alapjan, hogy ha a falat amit trekkelsz par frame-re megvilagitja egy auto azt a fenyiszonyvaltozast ne mintavaltozasnak vegye. Alapbol kikapcsolva kezdek, Az ilyenek altalaban megvezetik a trekkeket akarmiit is kapcsolok be.

Guided track: Nagyon jo ficsor. Mondjuk ritkan hasznalom, de akkor okkal es segit is. Kezi trekkeknel van egy parameter amivel „tracking guide”-ra lehet allitani egy trekket. Ha a tracking guide be van kapcsolva ezeket a tracking guide trekkeket figyelgeti, hogy hogyan ternek el tole az autotrekkek amiket talalt es ez alapjan hoz egy pluszz dontest a minta osszehasonlitason kivul a mozgasra is, hogy mi lehet jo pont es mi valoszinuleg nem. Sosem kapcsolom ki, ha nincs tracking guide track kizarasos alapon nem hasznalja. Ha csinalok, segiteni fog.

Consistency Check: Nagyon jo ficsor, szinte sosem hasznalom. Azt csinalja, hogy autotrekk utan atmegy az osszes ponton es leellenorzi a mozgasukat es ha olyat talal ami nem illik a kepbe kuka. Ebbol a szempontbol olyan mint a trekking guide, csak guide nelkul, matek alapjan - ami itt nem a solve, hanem valami mas advanced matek - dont ami eleg veszelyes. Ha kesobbi verziokban finomitjak nagyon jo lesz, most meg meg kell gondolni mikor lehet hasznalni. pl.: van 10 trekkem, mind jol meg 10 fram-en, de a 11.-en kimegy a kepbol amit trekkelnek es az osszes ugrik egyet mielott megszunne. Jo pont hijan az osszes mashogy fogja ugrani halaltusajaban az utolsot, ami semmilyen kameramozgasba nem lesz beillesztheto. Ilyenkor sajnos nem lamental, hogy akkor most jo-e az eleje, es a vege rossz vagy forditva, siman kidobja minden. Magyarul az osszes olyan trekkpontol meg fog szabaditani ami akar csak egy frame-en is ugrik a szazbol es ez azert igen nagy felesleges veszteseg tud lenni, azokrol a rossz trekkekrol nem is beszelve amik jonak tunnek, nem ugranak, pl 2 haz elenek a talalkozasat trekkelik es ugyis kezzel kell kihalaszni. Osszefoglalva egeszten nyugodtan hasznald olyan shotokon, ahol jok a fenyek, nincs rangatozo kamera, es varhato, hogy ha 500 trackpontot engedelyezel lesz is annyi. Ilyenkor siman donthetsz ugy, hogy autotrekk utani rosszpont guberalas helyett dobja ki ami nem teszik neki, hiaba dobja ki a jok mondjuk 10%-at, marad eleg. Nem amiatt hasznalom ritkan mert teveszt, hanem mert az oktatasi footage-eken kivul orultem ha 10 pontot talaltam, nemhogy kidobjam a sok jo 10 szazaleket ami nem fog hiányozni azzal az

elönnyel, hogy minimalizálni lehet utana a rossz pont keresést solve leott. Szóval ha a shotod engedi használ nyugodtan.

ezek a parameterek az autotrackekre vonatkoznak, kezi tracknel nincs mindegyiknek ertelme, de mivel a autotrackeknel es a kezi trackeknel a kozos parametereket is itt lehet beallitani erdemess ezt itt megtenni, hogy ne kelljen minden kezitracket egyesével beallitani amikor lerakod oket. A masik tipp a beallitasokra, hogy ranezesre nem fogod tudni hany pixels pattern es search area szukseges. Beallitasok elott siman lerakok egy user tracket, beallitom a search es a pattern areat es mivel ezeket az ertekeket lehet user tracknel is editálni egy ablakban, onnan kinezem az ertekeket, beirom a tracking parametersbe, kidobom a user tracket es kesz.

TREKKELES

USER TRACK

<CTRL> + <F3>

Az elo amit be szoktam allitani, hogy legyenek a trackek stabilizalva kozepre. Igy trekkeles kozben egybol latom felre ment-e: <ALT> + <n>. Ezutan keresek egy atlagos pontot es bealltom a parametereket – nem a trekkhez, hanem a track globalsban: <CTRL><F2>. Igy minden uj trekk amit lerakok ezekkel az alapertekkel indul.

Ha ez megvan lehet trekkeln. **Ha van a trekknek nehezebben lathato resze altalaban onnan kezdek**, pl motion blur, vagy mar nagyon kicsi es alig lathato. Jobban viselik a stresszt a trekkek ha a rossztol haladnak a jo fele mint forditva. Ha olyan van, hogy amikor elinditod a trekket meg egy kis potty, aztan mondjuk egy kereszt trekmarker lesz, neha elkezdi valamelyik szelet szeretni. Ilyenkor megis a jol lathato oldalon kezdek, es amikor latom, hogy elkezd csuszni valamerre ahogy csokken a pont vagy elmosodna megallitom, megkeresem az uccso jo framenet es elinditom onnan, esetleg finomitok a parametereken, mint pl. kisebb pattern area. Ilyenkor uj mintat vesz es nagyobb az esely a jo folytatrasra. Ha **minden kotel szakad egyesivel vegigpottyozom**, ami a gyors stabilizator miatt szinte nem macera mondjuk a shake-hez kepest. Mindezt persze csak akkor eroltetem, ha amugy nincs eleg track mashol, es varhatoan hianyozni fog az az egy, vagy pont jol jonne a lepcsozetessegezhez.

Ez eleg egyszeru, azonkul, hogy jol kell benzni a pontot es esetleg egy a kicsit parameterekkel kell babralnod ennyi a folyamat. Ha van egy jo trekked lehet jobb klick > tracking guide-ra rakni es akkor mar segitettel is egy kicsit az autotrekket.

TANACSOK:

- eloszor is a **solve-hoz nem kellenek olyan pontos trackek mint altalaban 2D re**. Itt ha berocog egy kicsit belefer, 20-30 pontbol a matek ki fogja szedni / atlagonli a recegesek egy reszet, pl.: ha az empire state bulidinget trekkeled es egy framen atugrasz a holdra valoszinuleg nem lesz gaz. Mondjuk ha mar kezi trekket csinalsz legalabb azok legyenek pontosak.
- ha ugy dantesz, hogy **tracking guide**-ot csinalsz egy kezi trekkbol, na az **alljon ugy, hogy elalszik rajta a legy!** bizonytalan frameken ne fojtasd, ha az osszes trekked popec ne allitsd mind guid-ra, inkabb egyet kettot a fobb feluletek, terek mozgasaibol.
- ha itt ott betakar valami hagyd ki nyugodtan azokon a frameken, de folytasd ha ujra feltunik a pont
- Ha uj mintaval elinditottad tobb framen, ellenorzeshez, hogy megtalaltad-e frame-eken at ugyanazt a pontot nagyon jo ugralni a kulcsframeken. ez a szogletes zarojelekkel megy
- track balra: „o”, jobbra „p”

- van, hogy amikor nem talalja a kovetkezo fram-en a mintat, vagy rosszat talal az segit valam miatt, ha semmi parametert nem allitok at, csak egyszerre csak egy fram-et trackelek. jobb klick a ponton > track one frame at a time.
- lesz olyan, hogy egy trekket az elejetol kezdesz el egyszer trekkeln, aztan a vegeteol mert ugy talalja meg jol. Ha kozepen a rossz reszeket ki kell dobni arra az a trukk, hogy kette kell vagni a trekket - jobb klick > split - kidobni az egyik elejet, a masik veget azutan ujra osszrakni oket. Ez azert ilyen korulmenyes, mer nem lehet pfTracken szakaszt kijelolni amit kidobsz. Vagy az adott framtol elore dob ki vagy hatra vagy az adott frame-rol. Ha akarmilyen trekknel a kozepre rossz sok framen mindig ez a szetvagom, kidobom kulon a rossz reszeket, osszeragasztom oket az egyszerubb.
- ha mar torles: ha elinditasz egy trekket, az adott frame keyframe lesz, akkor is ha nem igazitottal rajta. Ha ugy erzed, tudnal rajta javitani es elkezded valami miatt elolrol, akkor torold ki az elozo probalkozasokat, mert kulcsframek elott megall, azt ugy veszi, hogy tuti. Mivel a valoszinluleg allitottal a parametereken ujrakezdeskor, kicsit mashogy fog menni a trekk, mashol fog talalkozni a kulcsframeddel es lesz egy kics ugras
- ha valami miatt van benne egy kis ugras, de elotte es utana jo lehet probalkozni a jobb klick > overwrite trekk parameterrel es elinditani egyszer egyik aztan a masik oldalrol, de eleg ritkan sikerul fedesbe hozni ily trekkeket. A masik lehetoseg, es **ez eleg pontos, hogy jobb klick > show f-curves. Ez a trekk anim gorbeje amibe bele tudsz nyulni.** Siman kijelolod az ugras elotti / utani frameket, jol belenagyitasz a gorbebe, hogy finoman lehessen huzni, es kiszedesd a roccnest
- ez nem egetvero trukk, de jo, ne lepodj meg ha a pfTrack nem stabilizal le automatikusan egy pontot akarmennyire is kapcsolgatod ki be. Ilyenkor valasz ki valami mas es aztan ujra a pontot amit ellenorzol. Sajnos ilyen szokasai elfordulnak

AUTO TREKK:

Itt menet kozben nem sok beavatkozasi lehetoseged lesz, elore be kell loni a parametereket. Hogy ezzel ne menjen az ido, es ne 10 perc es 100 frame bambulas utan deruljon ki, hogy rosszak, ugy szoktam, hogy **keresek egy kritikusabb szakaszt, ott lelimitalom a timeRangget 10-20 frame-re es letrekkelem egy ket beallitassal. Ha megvannak a legjobb ertekek azzal trekkelem le csak az egeszet.** Lehetne azt is, hogy elinditod es megallitod ha rossznak tunik, de ilyenkor nem tartja meg a trekkeket, nem tudsz kicsit sem rajuk nezni majd, azokrol a parameterekkel nem is beszelve ami trekk utan futnak csak le. pl.: consistency check. A min track length parametert ilyenkor vedd le 5 ala, hogy ilyenkor azert megtartsa a rovid trekkeket is, de ne felejtsd el visszaallitani magasabb ertekre amikor a teljes footaget trekkeled.

ha ez megvan, kinyitot a timlinet es mehet a trekk. Ilyenkor az elso kerdese az lesz, hogy ujakat akarsz-e berakni es ezeket kidobja vagy folytassa ezeket. Teszt utan mindig ujkat akarsz berakni biztos ami biztos.

Ezzel a kerdeszel egy eleg jo lehetoseghez jutottunk. Ugyanugy mint user tracknel, autorekknel is van lehetoseg arra, hogy menet kozben valtoztass parametereket pl.: kisebb/nagyob pattern area, pl.: ott ahol beindul a kamera es van 10 frame ami nagyon motion blur-os, varhatoan sokkal nagyobb pattern area kell oda mint a tobbi alig mozgo szakaszon. Ezt ugy tudod elerni, hogy ha pfTrackben lelimitalod a timlent, trekkelsz aztan novelesz a timelineon es megint trekkelnel mindig fel fogja dobni a lehetoseget, hogy ujak keresese helyett a regieket folytassa. Az ilyen folytatas elott azt csinalsz a parameterekkel amit akarsz es az addott szakaszra szukseges. Egy hiba szokott rendszeresen elfordulni: Ha tul rossz a kep es nem sikerul meghosszabitania a trekkeket, akarmit csinalsz, hogy tovabb menj, vagy ujkat tegyel be beragad, es mindig azt fogja kerdezni, hogy kidobja-e ami van, fagy folytassa. Ha ilyenkor mar kicsikartal belole par jo trekket es meg akarod tartani oket, ugy tudod ezt kikerulni, hogy csinalsz egy uj motion groupot – ezekrol kesobb – es abba trekkelsz, oda engedni fogja. Ha kesz, siman atmanageled az eredeti grupba az ujbol a trekkeket, kikukazod a befejezetleneket es megmaradnak a jok es az ujak.

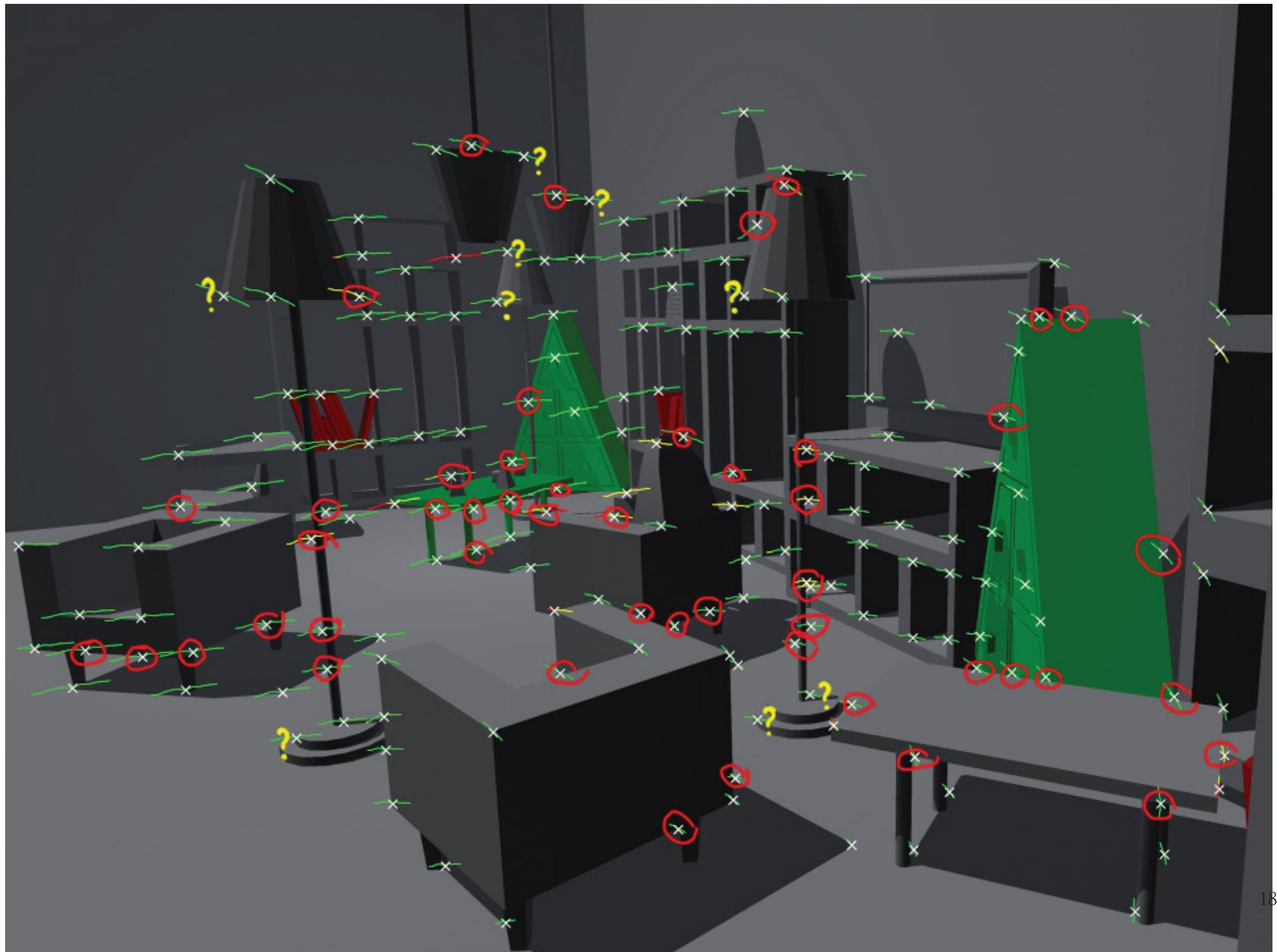
Olyan is lesz, hogy a footagen nem idoben, hanem teruletileg lesznek parameterezesi szempontbol eluto reszek. Ilyenkor Siman csinalj egy draft – tenyleg draft, a draftnal is draftabb – rotot, forditsd meg (inverze) es egy bizonyos bealltiasoknal trekkeld le azt a reszt. Aztan kapcsold ki / forditsd vissza a maszkot / csinalj mashova ujat, es trekkeld le a tobbi feluletet uj parameterekkel. Ezt is akarhanyiszor megcsinalhatod. Ezekkel a trukkokkel eleg jo ossze lehet vakarni kello szamu eleg pontos trekket, amit aztan szinte kukazni sem kell.

VADASZAT JO PONTOKRA:

Jo pontokhoz konneny lehet jutni kezi trekken es sima autotrekken tull egy ket trukkel. A legfontosabb jo paremeterekrol es a nem idoben hanem egy framen eltero minosegu pontokrol mar volt szo maszkokkal. Jo trukk szokott az meg lenni, hogy csak a legjobb pontokat autotrekkeld, hogy szigoritod a track parametereket ameddig lehet, levezed a max pontok szamat viszonylag alacsonyra – pl.: 20-30 – es letrekkeled. A thresholdot is erdemles ilyenkor magasra venni, hogy tenyleg csak a legjobbak legyenek meg. Ha ez meg van, kicsit atparameterezed, de meg mindig keves max ponttal. magas thresholdal, de mondjuk kicsit nagyobb pattern areaval es megint letrekkeled, ezt 2x, 3x eljatszva nagyon jo minosegu pontjaid lehetnek. – a lenyeg, hogy ahol egyszer mar talalt pontot a kovetkezo korben el szokta kerulni, ily ha 3szor letrekkeled ugyanazokkal a beallitasokkal elvileg az osszes a legszogorubb feltetleknek megfelelo pont meglehet anelkul, hogy le kene venni a thresholdot es beengedni gagyibb talalatokat is csak hogy legyen elegendo trekk.

Talkozni fogsz olyannal is, hogy kezi trekknel nem talal meg egy pontot de autotrekkel igen. Szoktam azt csinalni, foleg nagyon rosszul trekkelheto sok mozgasos shotoknal, hogy letrekkelem teljesen laza szabalyokkal egy csoporttal az egeszet. Ha ez megvan, kijelolok minden pontot es nem kidobom oket, hanem kikapcsolom oket a solve-bol. Jobb kliikk egy ponton, „ignore during camera solve” igy megmaradnak a jo / rossz pontjaim, de nem szamolodnak solvenal. A laza szabalyok miatt rendszerint tomenytelen hosszabb rovidebb pontosabb pontatlanabb pontom lesz. Ezekbol ugy lesz megis trekk, hogy szepen egyesvelet elkezdem oket ellenorizni, es amit jonak itelek visszateszem a solve-ba. A laza szabalyok nem azt fogjak jelenteni, hogy rossz trekkeket talal, hanem a jok mellet rosszakat is talahat es meg is tarthatja oket, mert seumi szabalyt nem allitok be, hogy magatol kidobjon barmit is. Igy a trekkek sokasagabol altalaban ossze lehet guberalni 10-20 pontosat, amiket meg talan ossze is lehet kotni hosszabb trekke.

Lesznek footagek, hol sok jol trekkelheto object lesz egymas mogott-elott takarasban. Ilyenkor lesz egy csoport jo trekked, de mindegyik rovid lesz, es amikor ujra felbukkan ugyanaz a targy kesobb legtobbszor tobb frame szunet utana mar nem tudja osszekotni a pfTrack. Lepcozeses trekkek eleresezhez ilyenkor segit, es ezt ajanlatos megtenni, hogy legalabb a fontosabb



helyeken levo jobb trekkek egy reszet, ha nem is az osszeset osszefuzod 1 trekke - nem baj ha leszek benne szunetek, csak vegig ugyanarra a pontra mutassanak. Ha ezt nem teszed meg azzal konnyen bukod a lepcsozetesseget, de ami megnagyobb baj es csusztat is a solve-on, hogy ha ugyanarra a terbeli pontra a sok rovidebb trekk miatt mas mas trekkekbol fog trebeli elhelyezkdest szamolni, amitol pontatlanabb lesz a kamera, es nehezebb lesz az illesztes. Osszefuzni pontokat ugy tudsz, hogy kijelolod oket, jobb klik az egyiken, es „merge tracks”

TRACK ELLENORZES

ittjon kepbe amit irtam a trekk parametereknel a max residualnal, a lepcsozetessegnel es altalaban a rossz pont guberalasrol.

Ha minden jol megy most van egy footaged egy csomo hosszabb rovidebb trekkel. A treck lengthben sorba rakva oket elvileg lepcsozetesek is, van mondjuk osszesen 600 trecked, framenkent 30-50 latszik, fremankent 10 jon be, 10 megy ki, atlagosan 10 frame hosszuak. Fasza. A pfTrack mutatja az utvonalkat, amit en azonnal kikapcsolok. Mert: akarmenyire is itel jonak vagy rossznak egy trekket egy matchMove szoftver, annak koze nem lesz a valosghoz, tehat attol a sok kis sarga-zold-piros csiktol csak latni nem fogsz semmit, ugy, hogy nem tudsz meg semmit. Az egész a matekra vonatkozik, ezeket szo nelkul hagyd figyelmen kivul. Mert: A kovetkezoket fogod latni: lesz egy csomo hibatlan trekked pirossal, lesz egy csomo ugralo trekked zolddel. Azokrol nem is beszelve, amik zoldekk is, le is kovetik is amire rafogtak, csak az eppen mozog. Vagy nem mozog, de ket egymas mogotti haz talalkozasa. Ezeket semmi matek nem fogja kidobni, Akarmennyit parameterezel az nem azert lesz, hogy ezektol megszabadulj, hanem, hogy eltalald azokat a beallitasokat amikkel a leheteto legtobb pontot minel hosszabban jol megtalalja.

Szoval ha ide eljutsz ekkor allsz neki ellenorizni. Felejtsd el, hogy egy 1 oras trekkeles 1234 pontjat atnezned frame-rol framre egyesivel. Ez 2 nap plussz. Itt egyedul annyit csinalj, hogy nagyjabol atfutod, es csak azokra koncentralsz, amik rossz pontra mutatnak.lasd fentebb good, bad, ugly pontok leirasat. Ha rafog egy kocsira siman dobd ki – ha nem parkol, ha hazak eleinek talalkozasat fogja, siman dobd ki. Ha teniszlaszti es jatszanak vele siman dobd ki, de ennel a resznel ne merulj el track szinten, mert ramegy a napod es van jobb megoldas. Irany a solve!

KAMERA

ha eddig nem allitottad be a kamerat - semmi baj, eddig nem is szamitott - most tudd meg, par szoban:

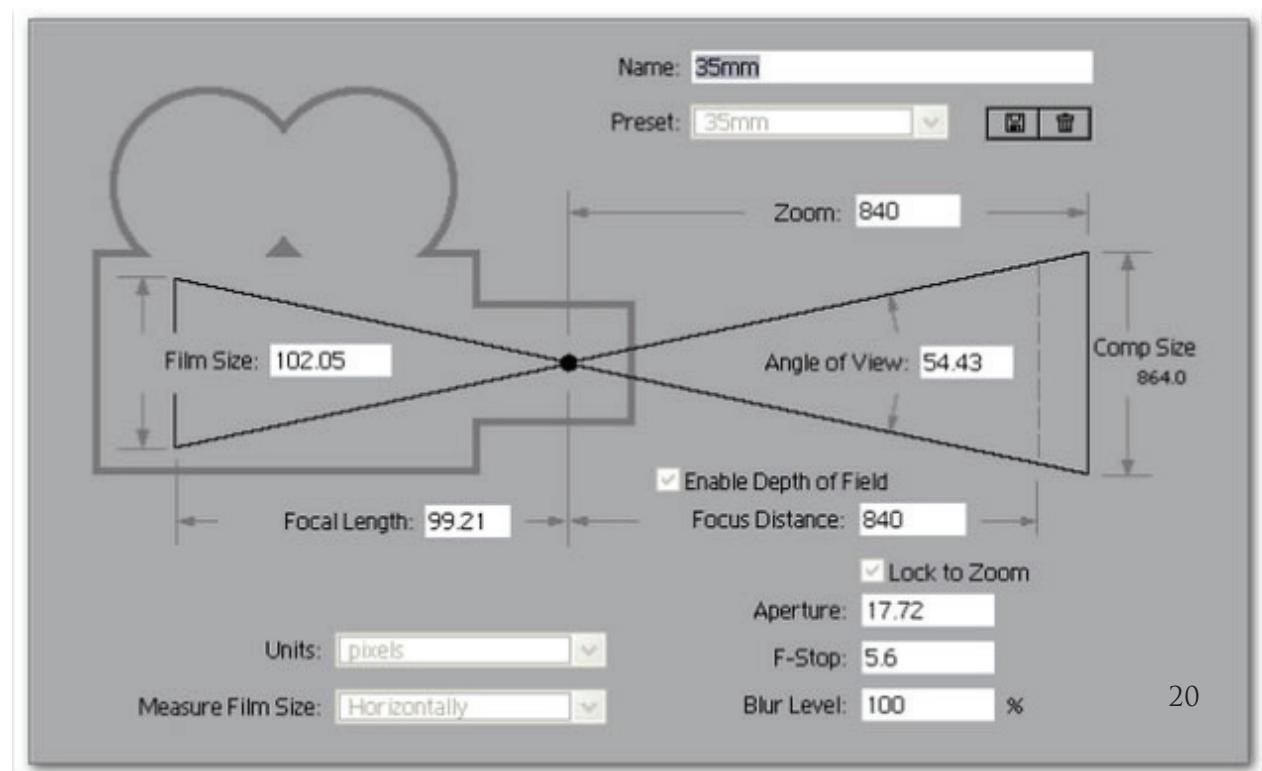
Eloszor is el kell dontened, hogy a trekkek, amikert eddig kuzdottel milyen mozgast irnak le. Ez azert fontos, mert van egy tipusu mozgas, ami nem is igazan mozgas, ezert mas solver mateket hasznalnak a szoftverek.

Ket fo mozgast kulomboztetunk meg: a csak forgas – **nodal pan** – es a barmilyen mozgast ami nem csak forgas – **free move**. Ez azert fontos mert a solve a perspektiva – parralax – valtozasain – parallax shift – alapulo mateket hasznal – ez az a bizonyos photogrammetria – , miszerint ha valami erre – arra mozog a kepen, az csak akkor lehet, ha a kamera arra – erre mozgott.

A nodal pan-t, ami szigoruan csak forgas a kamera fokuszpontja korul azert fontos megkulomboztetni a szoftvereknek, mert a csak elforgatott kamera helyzete terben nem valtozik a terben levo targyakhoz kepest, ezert nem is lesz perspektiva torzulas. Ha pedig nincs perspektiva torzulas, akkor nem lehet kamera mozgast szamolni, ami oke, de sajnos teret sem, ami nem oke. Nodal pan shotok solveja 3D nezetben tipikusan egy gombfeluletre szorja fel a pontokat, semmirol nem fogod tudni, hogy mi hol lesz, illetve az ellenorzes reszben hasznalt geometriakkal fogod tudni meghatarozni csak a dolgok helyet, de a pontokbol nem.

Az osszes tobbi esetben, ahol elmozdul a kamera, orulunk, lesz parallax shift, ami a matchMover baratja, abbol mar lehet szamolgnati minden.

Altalaban free move-nal lehet finomitani a mozgast, de mielott erre kiterek van egy dolgot ami sok bosszusagot fog okozni. Egy kamera csak akkor szamit matekileg allo kameranak, ha a forgas kozepointja a focus pont. Mivel a legtobb kamera ami csak forgonak tekintheto allvanyon van, ha azt elforgatják az elforgatas kozepontja meg veletlenul sem a focus pont kozepontja, hanem a kamera alatt az allvanyon valami forgathato mechanika. Ez valamivel odebbe lesz a focal ponthoz kepest, hatrebb es lejjebb, de sosem annyira, hogy elegendo perspektiva valtozast mutasson a footage egy komolyabb szamitashoz, altalaban ilyenkor ha latszik is parallax shift altalaban azon a hibahataron belul lesz ami a trekkekben benne van, ezert az



ilyen shotokkal eleg nehezen boldogulak a szoftverek. pl.: a tengyleges forgasi pont es a focal pont kozott van mondjuk 40 cm. Ha 90 fokban elforgatják a kamerát, akkor emozdult mondjuk 20 centit. Ha epp new yorkot forgatják, a több száz meteres tavolságok miatt az a 20 centi elmozdulás max 1-2 pixeles parallaxra lesz eleg, annyi csuszás pedig nehá a trekkekben is van, nem is beszélve a lencsetörzszáról, amit 100%-osan kiszedni szinte nem is lehet. Ilyenkor tipikusan kapsz olyan solveokat, amik 3D-ben ugy néznek ki, hogy az összes pont el van nyújtva a végénben. Ha ilyen van, jobb nodal pan-kent megoldani, az is kamu valamennyire, de ugy statisztikaval a solve kihuzza a hibákat és ad egy eleg jó eredményt.

A másik lehetőség ami pfTrackben ezeken atlendíthet, hogy finomhangolod a mozgást. Ez csak akkor lehet, ha a kamerában free move-ot állítottál be. Ilyenkor a „segment motion types” - <CTRL> + <q> - beállításokban választhatsz több lehetőséget közül, amik kicsit szigorítanak a solver matekon ami pontosítja a solve-ot. Az ilyen nodal pan, de mégsem shotoknál nehá minimalis parallax shiftel is dogos solve-ot kaphatsz ha „rotation, small translation”-re állítod a kamerára jellemző mozgást.

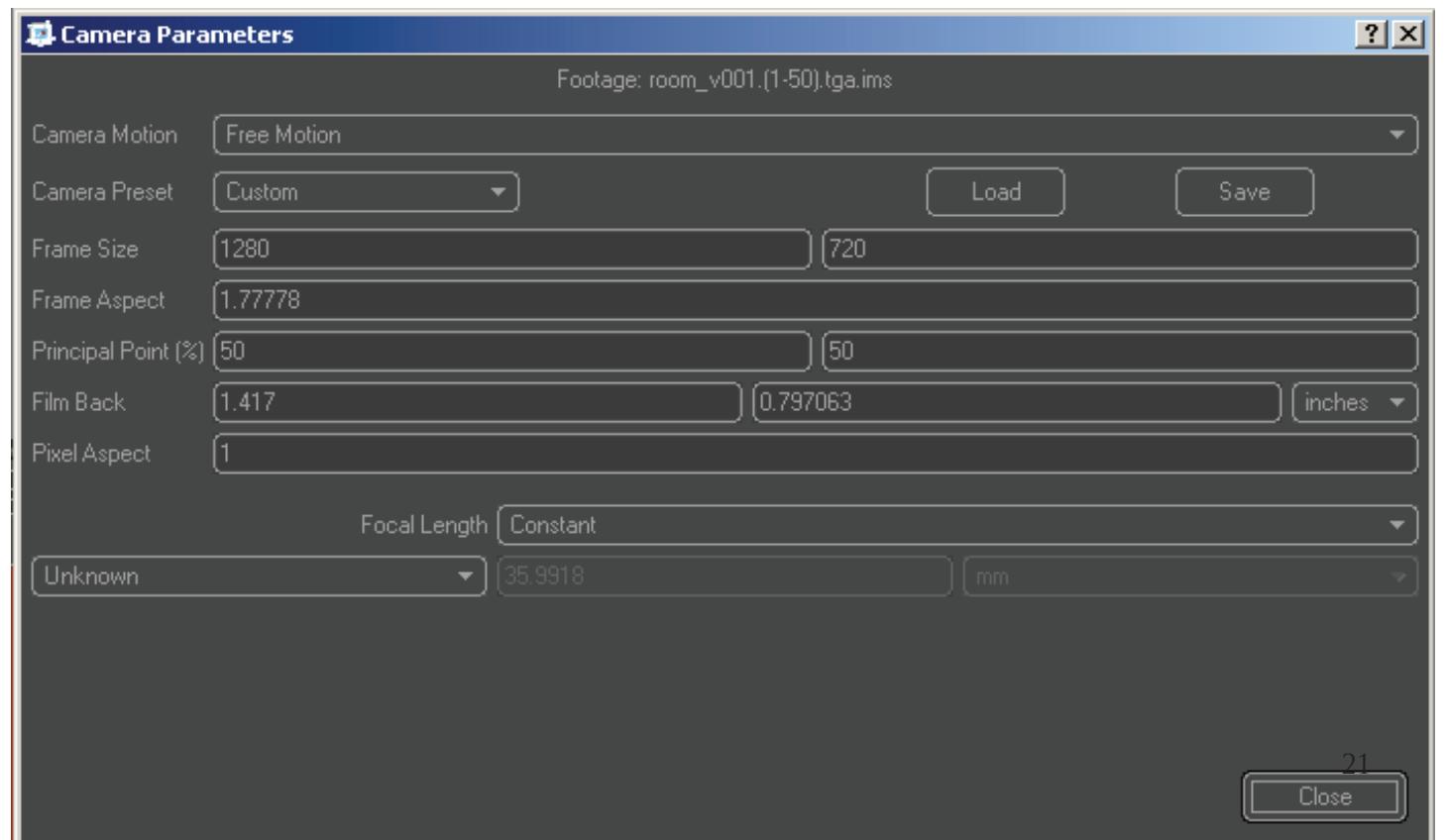
EZEKROL A PARAMETEREKRÖL NEHANY TIPP.

Meg mielőtt vadul nodal pan-re állítod a kamerát, nyugodtan hagyd free move-on akkor is, ha egyertelműen nodal pan, ugyanis free move-nal elérhetőek azok a paraméterek amikkel fel tudod vezetni a solvernek, milyen mozgásokra koncentráljon, és a lehetőség között itt is ott van a rotation only – ergo nodal pan. Ha az utóbbit paraméterek között állítod be, személy szerint soha nem találtam kulombseget a „camera parameters” es a „segment motion types” beállítások között, viszont ha az utóbbit használod, megszokod, elkezded használni és ha kitapasztalod szemre mikor mit érdemes limitálni, sokat segíthetsz a mateknak.

A MOZGASOKAT LEIRO PARAMETEREK NAGYJABOL EGYERTELMEUK:

- translation, rotation

minden mozgas
es forgas
felismerheto
a kepen - pl.:



leguritom a hegyen a kamerat

- translation only	csak mozgas - itt nem engedsz meg forgast, de a kamera barhogy mozoghat - pl.: a kamerat szep lassan eltoljak a Louvre kepei elott egy sinen, neha le-fel emelik ha magassabban log egy shagall, de egyszer el nem forgatják
- linear translation	ugyanaz mint az elobb, de nem megy le fel. semmi forgas, csak engyenes vonalu - de nem feltetlenul egyenletes - mozgas vegig.
- planar translation, rotation	mozog is forog is a kemera, de a mozgas egy sikban van, tehát csak 2 dimenzióban tologatják
- rotation only	csak es kizarolag forog a kamera
- rotation small translation	nodal pan, de nem a focal pont korul, tipikusan allvanyon levo csak forgo kameraknal jo
- interpolate	barmi, de nem szamoltatod a pfTrackel, mert nincs track. erről kesobb

Ezek a mozgasbeallitasok sokat tudnak segíteni, de mivel nagyon nehez eldönteni egy footagen, hogy liearis volt-e a mozgas, tenyleg csak egy sikban mozgatta-e a kamerat csak ritkán használom oket. Ami tenyleg szokott segíteni az a rotation small translation a fent említette okok miatt. Egyebkent is barmi szigorítás amit bevezetsz, hogy segíts a solve-nak, gyakrabban csap vissza rosszul mint segít, erről bovebben a Constrain-eknel irok majd.

Itt meg amit erdemessé tudni, hogy a footage-et fel lehet osztani mozgási szakaszokra a „dope sheet”en. Ez arra jo vagy nem jo, hogy ha az első száz frame-en csak forog a kamera az utolsón pedig csak mozog, akkor egy keyframmel 2 részre tudod valasztani a shotot és mindenket részre más mozgási parametert tudsz beállítani – jelen esetben a rotation only-t az elejere, és translation only-t a végere.

Az interplate is itt jön ki, ami onmagában nem mozgást jelent. Lesz néha olyan shotod, ahol valahol hirtelen megrandul a kamera, – mondjuk a best boy veletlenül belerug az allványba amikor viszi a turmixot lancfureszes gyilkosnak – a pontok eltunnek, bizonytalanok lesznek és minden elakad a solve azon a tíz frame-en. Ha ilyen van akkor általában a jobb részekre csinal az ember solve-ot és összebindzsizi Mayában. Itt segíthet, ha a rossz szakasz levalasztod, interpolációval. Ilyenkor a solve nem foglalkozik azzal a részkel, kihagyja, és megy tovább oda ahol mégint ertelmezhető mozgást mutatott neki a trekkekkel. Ez a mayában bindzsással annyival jobb, hogy minden egy solveban tartja a szemét, a két rész ugyanabban a terben lesz elhelyezve, nem kell azzal foglalkoznod, ha az eleje nagyob mint a vége vagy fordítva. Ilyenkor beigazítod mindenket veget mayában a proxy geo-hoz a közepű részen megkiegeszítet par kulcsal az animációt, de igazgatasról kesobb.

A SOLVE

korábban írtam már röla a jó trackek / lepcsozetesség résznel, nem is nagyon van mit tudni röla most. Eleg szigorú a matek, elvileg nem is nagyon lehet parameterezni, illetve a trekkeles maga a parameterezése, azt kell jól megtanulni hogy lehet belőle kiszedni a maximumot. Innen előbb annyi beleszolasod lesz, hogy „számláljon kamerát” illetve „tokre pontosan számoljon kamerát”. Ez a doski sem tul bubeszedű. Annyit mindenkepp érdemes szem előtt tartani, hogy a matchMove szoftverek 2 fele módon használják a trekkeket. A kezi, user trackeket darabra szigorúan berakják az egyenletbe, az auto trekkeket megszurik egy statisztikán alapuló matekkal. Ezt azért jó tudni, mert ha nem jön a user trekkjeid, hiaba vannak általában sokkal kevesebben simán megvezetik a solveot. Ha van elegendo auto trekk, eloszor mindig simán csak azokkal szoktam solveolni, ha van egy ket guide track vagy itt ott egy trekk általában jó kihagyni a solve-bol. A másik lehetőség, hogy soft constrain-re állítod oket. Ez azt jelenti, hogy szölsz a pfTracknek, hogy ha akarja, mert nem jönnek ki matekkel hagyja ki oket. Ezt mar csak azért is használom ritkán amikor kevés az autorekk és a kezi trekkeket is használom, mert ha nem szolok hogy kihagyhatja oket is simán kihagyja oket ha olyan van. Ennél jobb trukk, ha ugy szaporítod fel az autotrekkeket - merthogy ott jól jön a statisztika -, hogy atkonvertálod a kezi trekkeket auto trekke. Ez alapbol nem feature pfTrackben, ugy lehet megkerülni, hogy exportálod a kezi trekkeket, aztán importálod oket. Importál pedig minden megkerdezi, hogy autorekket akarsz vagy user trekket az újakból. Valassz autotrekket és maradj abban a korben vannak ahol a matek miatt több egyenetlenség belefér és jobban össze is dolgoznak azokkal a trekkekkel amik tényleg auto trekknek készültek.

Ha nagyjából kidobtad a ranzesre rosszakat, johet az eső solve. Ez nem a vegleges lesz, ezzel fogod letisztazni a rossz trekkeket.

Indíts el egy solve-ot, csak auto trekkekkel <CTRL> + <F7>. Varhatoan par perc lesz és meg elegendő eredményt ad. 3D nézetben megnezheted, de még össze vissza lesz. Ez a solveot, illetve egy ket ilyet fogsz használni kukaszra. A pfTrackben alul solve után aktiválva valik a ket tab: projection error, alias proj-e és camera error, alias cam-e. A camera error ami megmutatja milyen a solve - plussz a statisztika -, de ott még nem tartunk, kukaszánál most a projection errorról nezd.

Amikor megvan egy solve, kb 8 tizedesjegy pontossaggal berakja a kapott terben a trekkekből számolt pontokat. Ezek fixek, nem rocnak, meg se nyekkennek. Ha ezeket a kamerán át visszavetít a footag-ra, megkapja, hol lenne annak a pontnak a trekkje. Azért csak lenne, mert sosem lesz

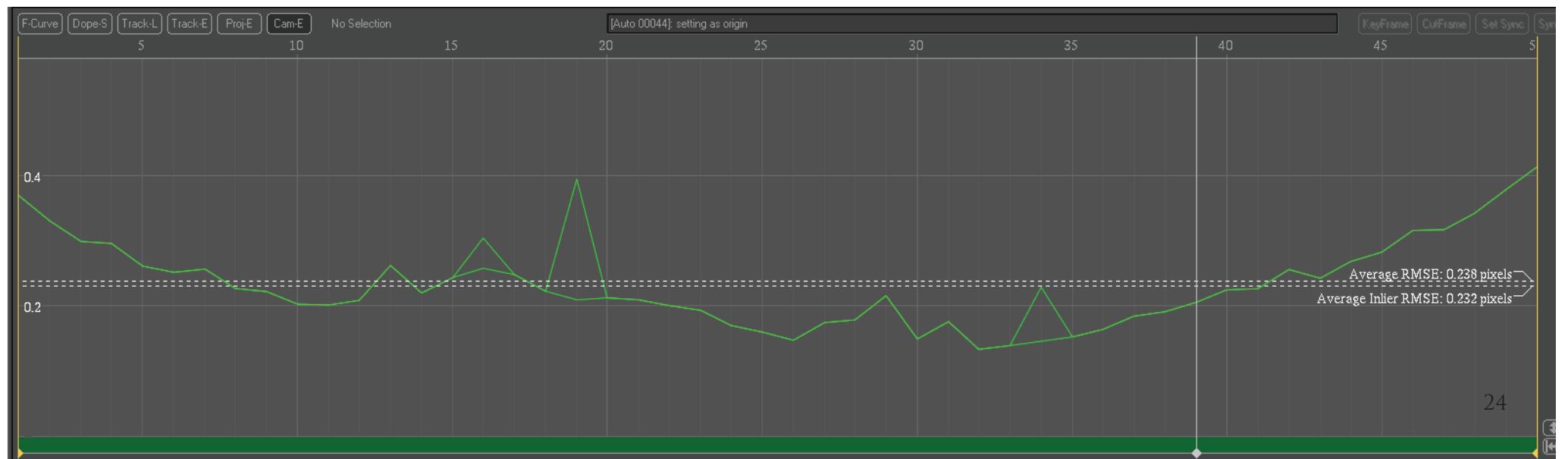
Frame	Total Features	Inlying Features	Avg. Inlier Error	Avg. Error	Max Error
16	87	98.85%	0.258396	0.304753	1.52905
34	87	98.85%	0.146686	0.229515	1.65302
19	89	98.88%	0.210121	0.394888	3.16118
1	77	100.00%	0.369676	0.369676	1.06296
2	77	100.00%	0.33074	0.33074	1.01455
3	77	100.00%	0.299446	0.299446	0.892769
4	77	100.00%	0.296426	0.296426	0.90366
5	77	100.00%	0.261745	0.261745	0.802438
6	80	100.00%	0.252288	0.252288	0.714997
7	80	100.00%	0.257122	0.257122	0.910742
8	80	100.00%	0.22749	0.22749	0.628241
9	80	100.00%	0.222714	0.222714	0.606489
10	80	100.00%	0.203386	0.203386	0.525833
11	80	100.00%	0.202223	0.202223	0.631574
12	81	100.00%	0.209269	0.209269	0.720673
13	84	100.00%	0.262749	0.262749	0.901879
14	84	100.00%	0.220564	0.220564	0.730385
15	85	100.00%	0.244222	0.244222	1.11686
17	88	100.00%	0.248725	0.248725	1.16189
18	90	100.00%	0.223275	0.223275	0.933271
20	92	100.00%	0.213129	0.213129	0.864217
21	93	100.00%	0.210208	0.210208	0.826025
22	93	100.00%	0.201193	0.201193	0.865181
23	93	100.00%	0.19403	0.19403	0.858076

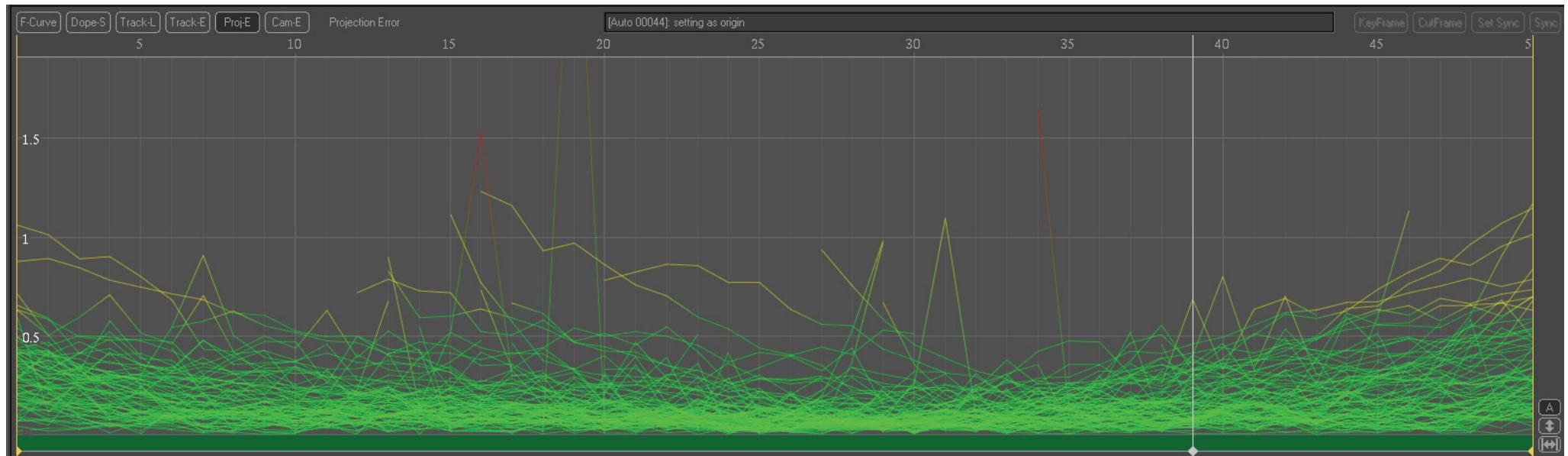
olyan pontod ami ugyanoda fog esni mint a trekk amibol szmolva lett. A legfontosabb trekk is rocog kicsit, el-elmaszik az anitaliason, blur-on, barmin, de ez a pontbol visszavetitett track nem. A projection error azt mutatja, hogy az adott trekk az adott frame-en mennyire csuszik el attol a visszvetitett trekktol ami abbol a pontbol lett vetite ami a trekkbol készült. Igy minden egyes pontra minden frame-en meg tudod nezni, hogy hany pixelt csuszik a belole szamolt trekkhez kepest, es ha hibahataron felül van vissza tudod nezni, hogy jol szamolta-e a solve a 3D-s pontot vagy sem.

A hibahatarok a következők: ha a legtöbb trekked vegig 0.3 pixel alatt csuszik, majdnem biztosan jo a solve, ha ilyenkor nem azt latod 3D nezetben amit akartal, a focal length-el lehet a baj, esetleg hiaba jo a legtöbb pont a tul sok mozgas miatt kevesek bizonyultak pontos solve-hoz. Ha a legtöbb pontod 0.6 pixel alatt van meg boven lehet jo, ilyenkor azonban mar lehet, hogy egy ket frame atcsuszit hibahatar fole, ugyanis sosem lesznek egyenletesen jok a trekkek. 1 pixel alatt valoszinuleg lehet pontositani, de meg akar lehet is jo, csak nem kukaztal eleget es csak a statisztikad rossz. de 1 pixel folott szinte biztos, hogy valamit ki kell talalnod.

Es az ellenorzesre visszaterve: megvan a solve, megvannak a hibahatarok. Ilyenkor azt csinald, hogy menj szepen vegig a nagyon kilogo trekkeken, jelold ki oket alul, ahol mar nagy az elozdulasa. Ekkor valts at trekk nezetbe <ALT> + <Y>. Ilyenkor nem a 3D-s pontokat, hanem a trekkeket latod amikbol készultek. Stabilizald le amit kijeloltel es nezd meg jogos-e, hogy olyan sok lehetett a csuszasa. Altalaban jogos lesz, foleg elso korban. Olyasmiket fogsz talalni, amiket siman talalomra keresgelessel szinte biztos nem, es ez az ami miatt nem szabad sokat tokolni az autotrekk utan egybol a guberalassal. pl.: vegig jo a trekk, de egy fram-en elrepül előtte a legy es ugrik egy pixelt, jo a trekk, de a szelen kimegy amit fog, es az utolso utani frame-en benmarad meg egy ugrasra, jo a trekk, de nem valos pontot mutat, hanem 2 egymastol 2 fenyevre levo haz oldalainak metszetet trekkeli, megsem doglott a pöttöös macska a bokorban, hanem szuszog es remeg a trekk. A latszolag szelcsendes időben mozdulatlan fa pedig neha tornadot tud podukalni az ossze vissza lassu kis himbalozasaval a trekkekben, stb...

ha megvan a pont es tenyleg rossz azt kell tenned amire gondolsz. kidobod a rossz reszen. Ha 3-5 frame marad jo es van meg eleg siman kukazd az egeszet. Az van, hogy solve utan a





pfTrack nem engedi majd használni a remove menüpontokat. Ez valószínűleg azért van, hogy megse editalj olyan trekket amiből epp egy solve-ot ellenorzi. Mielőtt egyesével törölne a solve-ot minden rossz trekk javítása előtt, direkt vagy veletlenül, de működik a kidobás shortcuttal. **<SHIFT> + <><>** kidobja az adott framot előre, **<SHIFT><>>** kidobja az adott framot hátra. Ez ezzel ott tartasz, hogy tul sok guberalas helyett, csak simán az értékeket nezve szépen egyesével vegig tudsz menni az összes rossz trekken egy proba solve-al. Ha ezután atmész a camera error-ra, most fogsz latni mint az elejen. Eloszor is, a két gorbet fogsz latni, az also azoknak a pontoknak az atlaga hibáitól belül esnek, a másik az összes pont atlaga. Ezt a hibahatart be lehet ugyan allítani a preferencoakban, de azonkívül, hogy mashol huzza meg a nem találtam semmi hasznát.

Tehát most kukaztal rossz pontokat rossz frameken, és megnezed a projection error-t. Azt fogod latni, hogy az két görbe elkezdett közelíteni egymáshoz. Ez jo, ugyis az a cél, hogy a legtöbb pont hibahatáról belül legyen, szóval ha ez történt jo az irány.

Ha atmészted a legrosszabb pontokat mondjuk 1 pixels csuszásig a 2 gorbed majdnem fedesben van, de ezzel nincs vége. Most csak annyi történt, hogy a solve nem lett pontosabb, csak a trekkek, simán csak kidobtad azokat a pontokat/szakaszokat amik a rossz statisztikát adtak jogosan, de ekozben nem számolódott uj solve. Most egész egyszerűen dobda ki a solve-ot és csinald meg megegyeszer.

Ez mar sokkal pontosabb lesz megha elsőre nem is így fog tunni. Az fogod latni, hogy a camera errorban megint felugrott legalább az egyik görbe, a projection errorról meg jobb nem is beszélni. De ez nem baj. Simán csak annyi történt, hogy az uj solve-nal, a javított trekkek miatt attrendeződtek a ero / súlyozás viszonyai a trekkeknek, és mas hibák jöttek ki, de mindenkeppen kisebbek és kevesebb, megha esetleg ílyesztő is lesz a sok piros csík latvanya.

Most simán csinald meg megegyeszer ezt a kukazast. aztan megegyeszer és megegyeszer...

Az a tapasztalat, hogy 3-nal többször már nincs ertelme, szép lassan elkezdenek azok a pontok felszaporodni, amiket jóval ittelsz rossz eredmennnyel és nem dobod ki mert lehet hogy a többi sok rossz pont vezette meg a solveot és ez a jo. Lesznek ilyenek. A 2., 3. kor vegyre simán le lehet vinni a pontok többségét 1 pixels csuszás alá, hogy csak 1-2 fram-en ugranak fel, és

a camera error 2 gorbeje is nagyjabol fedesbe kerul. Nezd meg 3D-ben mit mutat. Ha azt amit vartal elvileg kesz a jo solve-od, lehet ellenorizni. troubleshootrol kesobb

CONSTRAINTS

csucstechnika, sokan szeretik, szerintem kamu, en sosem hasznalom, es batoritanalak, hogy te sem tudd, de ha ugy erzed segit nyugodtan.

MatchMove szoftverekben lehetoseged szokott lenni arra, hogy pontok egy reszere szabalyokat allits fel, miszerint:

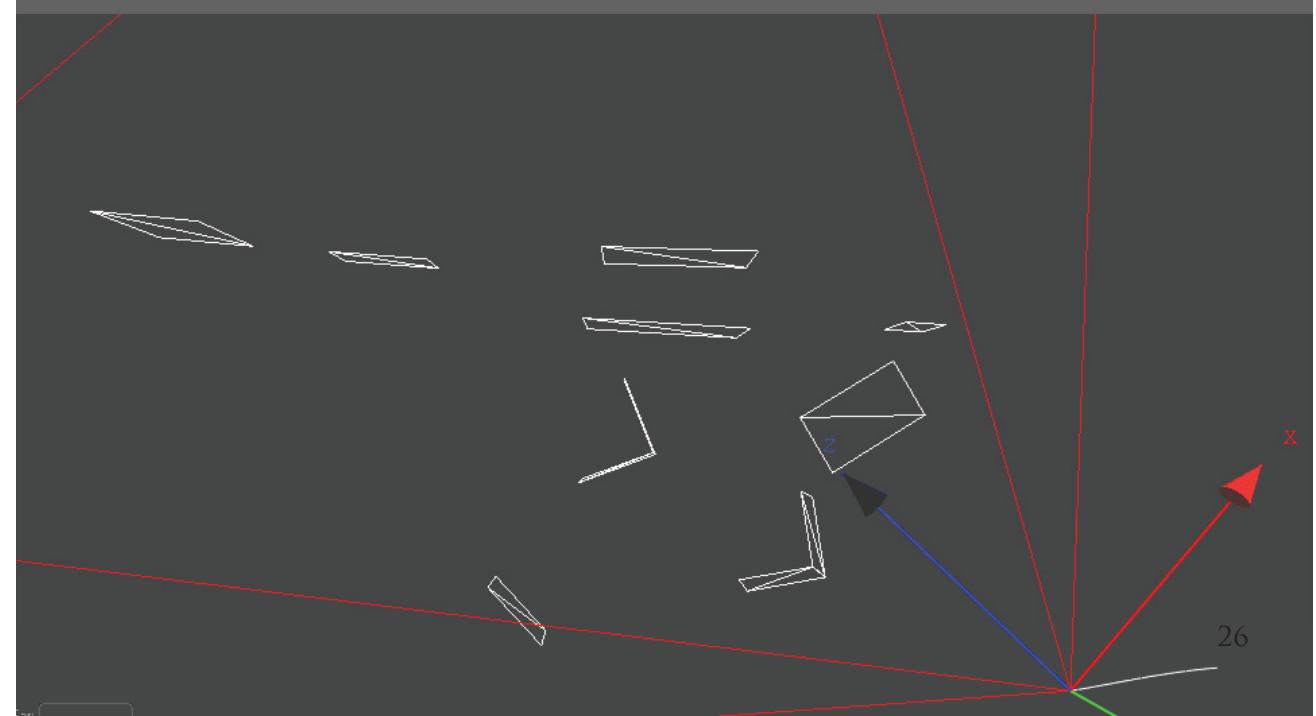
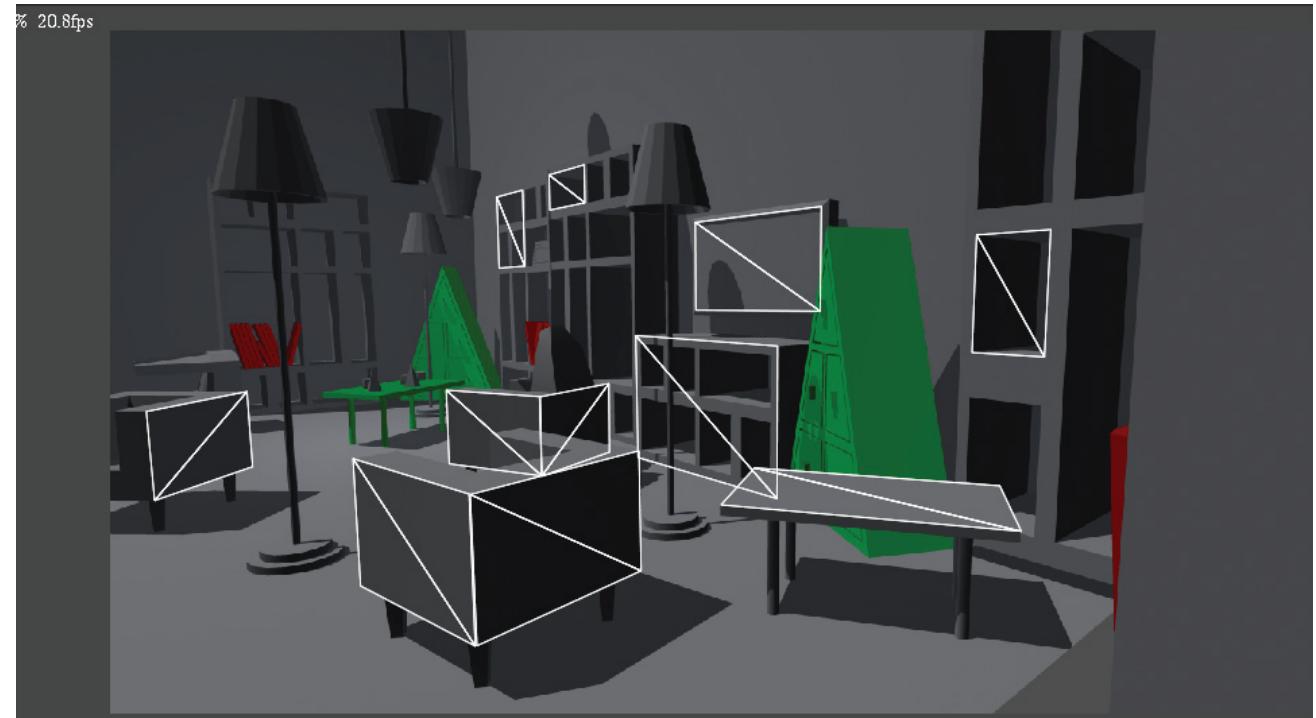
- ez es ez es ez a pont sikban van
- ez es ez es ez a pont egy vonalban van
- ezek a pontok, sikban vannak es parhuzamosak az xz sikkal,
- stb...

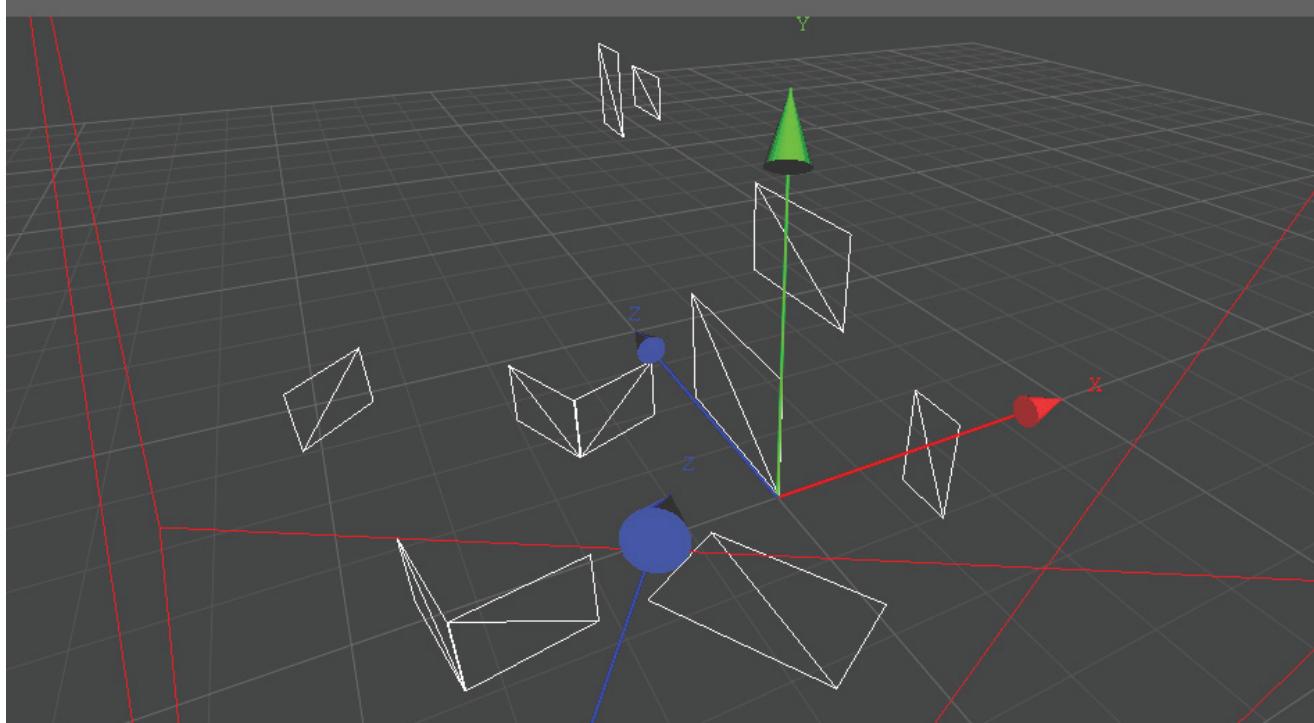
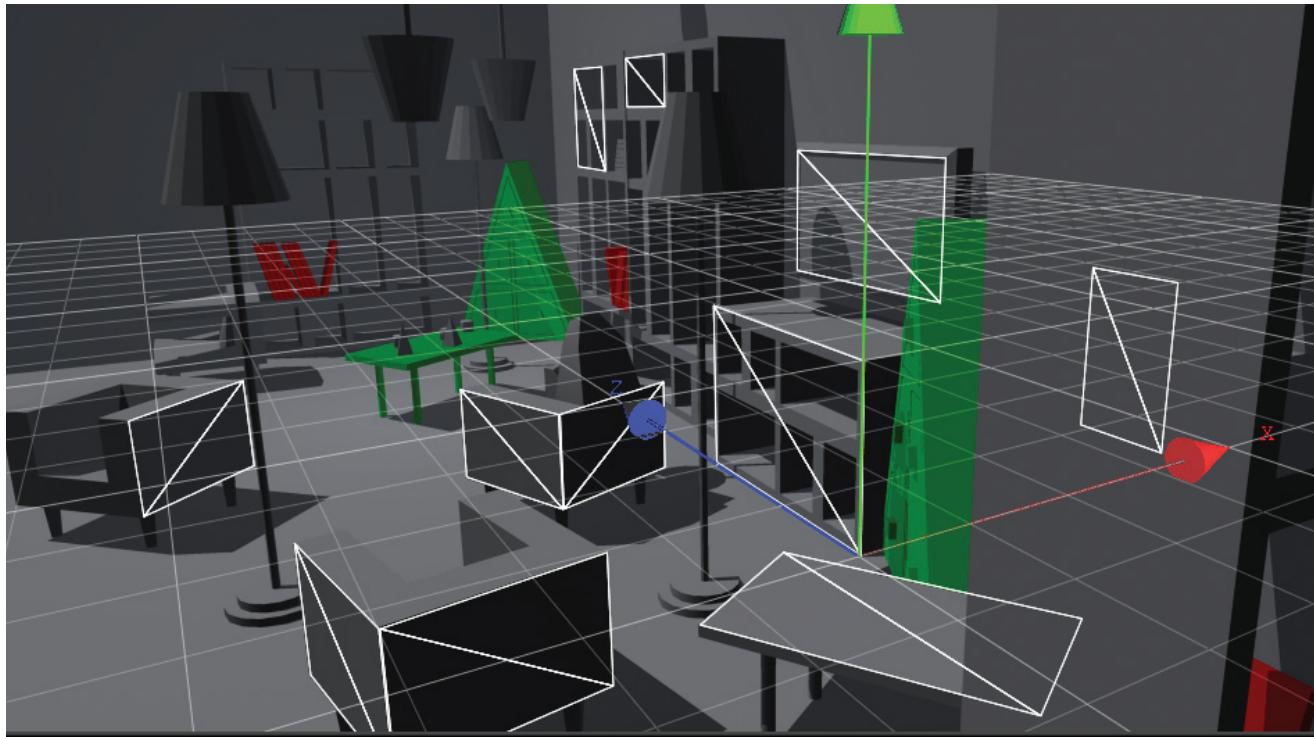
SOLVE ELLENORZES

Mielott atmesz mayaba amennyire lehet ellenorizz le pfTrackben a solve-ot, igy nem kell oda vissz ugralni. Ellenorzes kozben raadasul polygonogat fogsz hasznalni, amik majd az beigazitasnal direkt jol is jonnek mazaban.

Elo korben nezd meg, hogy nagyjabol felismered-e 3D nezetben a dolgokat amiket trekkeltel. Ha rosszak a focal length beallitasok, rosszul allitottad be a kamera mozgast, vagy nem puoltad le elege a trekkeket olyasmiket lehet latni, hogy szinte egy vonalban elnyujtja a pontokat. Neha direkt a kamera moe is tesz ami ugye vicces. az ilyen trouble shootrol kesobb. Ha ugy tunik sikerult a solve, nagyjabol megvannak az objectek amiket trekkeltel a legjobb ellenorzes ha teszt objecteket teszel be.

Eloszor is felejtsd el a katicabogarat, a gombat, a pfTrack telenet, stb... amiket matchMove szoftverekben tipikusan be





lehet tenni ellenorzesnek ugy, hogy kijelolsz egy pontot es leraksz oda egy test objectet. Ez kamu, siman megvezet. Ha passzol a pont az meg meg lehet vele nezni, de azt meg lehet nezni bambi nelkul is. Azt pedig, hogy jo helyen van-e nem mutatjak meg, ha leraksz egy ponthoz egy ilyet nana, hogy szepen menni fog vele, csak lehet hogy az a pont epp 2 tengeri merfolddel lesz hatrebb mint kene.

pfTrackben a legjobb ellenorzesi modszer ha polygonokat raksz be a modelling palettarol. Es ezek kozul is csak a negyzetet ajanolom - 2 fajta van, az egyik fel van osztva 10-10 reszre, ne ezt hasznald mert nem fogsz latni semmit, a masik siman egy quad, 2 haromszogbol. Az elleorzes ugy meggy, hogy keresel valamit amihez egy quad-ot vagy haromszogot hozza lehetne illeszteni, es leraksz egyet az elso fram-en aho mind a negy/harom sarka latszik. Ezutan megfogod a geo sarkait es odahuzod ahol a kepen lennie kene. Ezutan elmesz az utolso fram-re mielott kimenne a kepbol, es ott is raigazitod a megfelelo sarkait a megfelelo sarkokhoz. Meg kozenyen egy harmadik frame-en is megteheted a biztonsag kedveirt, de a folott folosleges.

Ez az egyetlen modszer, hogy olyan objectekkel ellenorizd a trekket, amiknek nincs semmi kozuk a solvehoz vagy a trekkekhez, siman kep alapjan igazgatod be oket, ezert ha nem jo a solve, az rogtan lebukik ezekkel ha nem sikerul beilleszteni valamit amit a kep alapjan be kene tudni. - kis tulzassal.

A modszer lenyege, hogy amikor elkezded igazgatni az objectet a pfTrack keres neki egy helyet, meretet es agy elforgast a terben, ahol rapasszol a kepre. Ilyenbol az elso lepesben van vegtelen-1 megoldas. Amikor egy masik framen

is elkezded beigazitani mar csak olyan helyeket tud/fog keresni, amikkel fent tudja tartani az a feltetelt amit az elozo frame-en beallitottal, miszerint azon a fram-en hogy is nez ki a geo. Ha ket frame keves, egy harmadikkal altalaban egeszen pontosan be lehet loni mi hol van, ilyenkor mar nagyon le vannak neki szukitve a lehetosegek, pici csuszas ha van benne az ettol kijon.

Ha igy be tudsz tenni legalabb 2-3 quad-ot, amik parhuzamosak a tengelyekkel (xy, xz, yz), vagy az iranyokkal amik a kepen vannak es azok vegig a helyukon maradnak lejatszas kozben, nem nagyon kotozkodhet majd a 3D-s akinek atadod a solve-ot.

GEO TRACKING

A trackpontos treckeles helyett lehetoseged van geometriaval trackelni. Ez azt jelenti, hogy fogsz egy lehetoleg pontos modellt, raigazitod a footag-re egy frame-en, elindítod a trekkelest es a pfTrack megprobalja megkeresni framerol framere azt a kameraallast ahol a geo passzol a footage-en levo objectre amire a referenciaframe-en raigazitottad. A trackpontok helyett itt a vertexek (?) fognak trekkelodni,es mivel a trackeles maga nem valtoztathat a geometrian, ugy keres megoldast a kameramozgasra, hogy figyelembe kell vennie, hogy egy vertex milyen kapcsolatban van a tobbivel - tavolabb/kozelebb, jobbra/balra, stb.... Ez a geometria szigoritasa miatt altalaban sokkal pontosabb eredményt ad es altalaban gyorsabban is. Teremszetesen nem beleszamolva ha neked kell modellezni is. Ha megis, jo trukk szokott lenni egy viszonylag jo sima track-el kezdeni, a trackpontokbol kesziteni egy poligon objectet, es azt ujra letrackelni. Ha mar egyszer le van treckelve jol, termesztesen nem kotelezo a geo track, annak csak a abban az setben van elonye, hogy a terben mozgo objectek mozgasat kell visszaszamolni, raadasul allo kameraval, ami mayaban lehetove teszi, hogy a matchMove kamerat a group-al egyutt barmi mas mozgo targyhoz - altalaban a fo kamera - illeszd.

Ha van geometriad, es fo kamerat kell trekkelni, a dolog egyertelmu: tracking geot importal, kamera parameterek beir, trackel, exportal. Ha egy object mozgasat kel trackelni tobb megoldas lehetsegese.

Lehet a fo mozgas mellett groupokban lettrackelni az objectet. Mivel a pfTrack soem fogja tudni csupan a kep alapjan, hogy kicsi, lassu, kozeli az object vagy tavoli, nagy es gyors, ezt kezzel kell beigazitani meretre track utan. Az ilyen esetekben konnyebb lehet az a megoldas, miszerint mindenhez kulon track keszul es mayaban kerulnek ossze. Sok gubanc nem lesz, ha igy csinalod:

- lettrackeled a fo kamerat kulon,
- lettrackeled az objectet/objecteket kulon-kulon, allo (stationary) kameraval.

a stationary kamerabeallitas azt jelenti, hogy nem a kameraban lesz a mozgas, hanem az object fog mozogni, es a kamera siman mozdulatlanul az origoban lesz. Ezt azert szeretjuk, mert innentol barmi mashoz hozza lehet kapcsolni az adott object mozgasat anelkul, hogy serulne a solve. Ha az object trackek megvannak, a pfTrack solve grupokat csak be kell parentolni, gruppolni, constrainelni a fo kamera ala. Mas munka nem is nagyon lesz vele, hiaba lett kulon trekkelve az egész, az objectek mozgasahoz hozzaadodik a fo kamera mozgasa is, ezert passzolni fognak ahoz a kamerahoz is. Termesztesen a gruppolas, parentolas, constrainelas elott a group origojat, ami megegyezik ezen esetekben az allo kamera helyevel, a fo kamerahoz kell ileszteni, hogy ne legyen eltolodas.

Ekkor meg a merettel lehetnek pontatlansagok, foleg ha nm volt elore legyartva meretre a traking geo, de sima scale-el nyugodtan be lehet illesztgetni minen egyes kuln trackelt pfTrack grupot.

AMIRE ODA KELL FIGYELNED:

tracking geo beemelesenel - a tracking geo OBJ - az also dolgod legyen, hogy a geon jobblikkel atallitod a a geot „right handed”-re. Volt egy programozo pixelfarmnal aki orult, hogy sikerult beleirnia ezt a funkciot es akarta is, hogy hasznald: a pfTrack alapbol minden bal kezes koordinata rendszerbe hoz be, - meg a sajat geometriait is - megha valoszinuleg az osszes szoftver alapertelmezesben jobb kezes koordinata rendeszert hasznal is. Onnan fogod latni, ha ezt elfelejtettek, hogy a tracking geo ugy fog kinezni, mintha ki lenne forditva - ami elol volt hatul lesz, ami balra volt jobbra lesz...

A masodik amit be kell allitani, az a stationary kamera. Ezt a beallitast eloszeretettel felejti el ha geo import elott lett beallitva. Nem baj, ha elfelejtod, egyszerulen siman nem fog elindulni a geo track azzal a hibaval, hogy nincs fo kamera solve, ami nelkul nem tud geo-t trackelni egy adott treben ami nem letzik. Utolag is korigalhato a dolog, csak agyzsibbaszto negyed ora geo igazitatas utan talalkozni a hibaval, mert utolagos atallitas utan termesztesen lenullazza a kamerat amitol elugrik a geo es kezdheted elolrol az also beallitast. Ha ilyen

van siman at kell allitani a kamerat pl.: free motion-re, leokezni, majd ujra vissza stationary camerara. Onnan fogod tudni mikor vette be. ha a geometria elugrik - gyakran a kepen kivulre - es a kamera alapertelmezett helyre kerul. Ez $x = 0$, $y = 10$, $z = 40$, a makera -z tengelyen nez, es meglatod a koordinata rendszert. Ez ellentmond annak amit korábbi irtam, hogy a kamera az origoban van. Ugy kerul oda, hogy a solve vegen a camera menuben, transform kamera-val odateszed: absolute transform, $xyz = 0,0,0$

- 3D blur visib./max

A motion blur es a sampling ertekek valtoztathatoak, a lenyeg a jo ellenorizhetoseg, illetve, hogy pl. 2 ora matchMove-ra ne 1 ora teszt render jusson. A fenti beallitasokat hasznalva 50 frame varhatoan 10 percen belul lefut.

pfTRACK PREFERENCES:

general tab:

- no spaces in filenames: ON

interface tab

- store image thumbnails: OFF
- guess frame boundary: OFF

performance tab:

- use memory cache: ON
- cache size (Mb): 750 - 1300 depending on RAM
- prefetch during playback: OFF

calibration tab

inlier threshold (in pixels): 0.6 (maximum)

MATCHMOVE DELIVERY TO DNEG:

maya matchMove scene delivery to Dneg:

maya binary

- camera should have no parents,
all animation in the world
coordinate system

- locked sensible data:
translation
rotation
focal length
camera aperture

backed in curves

jpeg seqence as image plane

no resolution or film gate

Dneg geo included

- naming convention:
maya scene: [shotname]_mm_[version] – ezt felirja az aktualis nevezektan, ha a
Dneg elfogadja
camera in maya scene: shotCamera_[shotName]_[pfTrack solve version] – ezt
felirja az aktualis nevezektan, ha a Dneg elfogadja

- maya scene preferences:
mesaurement: degree, cm, 24 fps

(>REGI MECMSOOVE DOKSI ELMKEZTETO<)

ALAPFOGALMAK:

a matchMove-ot kulonfele tutorialok mas mas szovegkonyezetben camera tracking-kent, kamera solve-kent illetve kamera calibration-kent is emlegetik.

kamerak tulajdonsagai amik a matchMovehoz kodtak:

film back / camera aperture: a kapu / ablak / luk amin keresztul a feny a filmet eri.

Kozvetlenul a film elott helyezkedik el, maszkkent funkcionál, ami ezen az ablakon kivül esik nem exponalodik.

focuspont: a gyüjtöptőn amin keresztül az optika a negatívhoz juttatja a fénnyt
focal length: a focuspont es a film back kozotti tavolság

angle of view: latoszog - az a szög amit a kamera a focusponton keresztul belát
photogrammetry: a 3D-s kamera mozgasat es paramtereit 2D-s kep

alapjan kiszamolni kepes matematika, a matchMove alapja

Lens distortion: lencse hiba, pl.: az egyenesek nem egyesek, jellemzően a kép széle felé nő a hiba, pl.: házak falai ferdek, villanyoszlopok ferdekk, és általában mindennek az éle ferde, amiről egyébként nagy biztonságával ki lehet jelenteni, hogy a valóságban egyenes.

mivel a CG kamerák torzításmentes képet adnak - torzításmentes, lencsenélküli pinhole

kamerák -, a forgatott anyag és a CG elem külömbsségeket mutat tökéletes matchMove esetén is.

Lencsetorzításbeli különböszük pl.: ha a mozgások, elhelyezkedések szemre stimmelnek, de a kép széle felé a digitális képi elem egyenletesen elmozdul az eredetiől. A lencsetorzítást minden esetben ki kell szedni a footage-ból matchMove előtt!

parallax: mozgas közben letrejövő prespektiva torzitas

nodal pan: kameramozgas, ahol a kamera csak forog a sajat kozepontja korul. A

kozepont ideali esetben a focuspont

free move: kameramozgas, ahol a kamera helye (is) valtozik

focal lenght - variable vs. fix

- feature track

- geometry tracking
- matchMove
- solve
- pan shot
- free move
- residual
- inLier
- false pozitiv
- search/pattern area

- MI A KULONBSEG 2D ES 3D TRACK

- miert nincs track offset?
- hany pont kell?
- zajos pontok, feature track, user track
- hard, soft track

KOZOTT?

- mertekegysegek
- camera beallitasok
- frame/sec
- track merge, split
- lepcsozeset track foltonossag
- szigorú beallitasokkal tobbszori trackeles
- parameterek a megengedobolt a szigorubb fele
pl.: user track alapbol soft
- track hossz (kb. 7-9 frame minimum, 4 frame allatt valoszinuleg nem
nagyony van ertelme)
- szelek maszkolas
- a lehető legkevesebb feature track vs a lehető legtobb pontos
feature track
- mask-al iranyitott feature track
- proxy / nem proxy
- track test limitalt frame range-el
- feature track kukazas:
 - egyesevvel ki kell valogatni a rossz trackeket.
 - view lock es lejatszas.
 - delete before, after
 - ha a kozepre rossz, trak split -> kukazas -> track
merge
- residual:
 - nagyony nem erdemess nezni: false pozitiv
- koordinata rendszer lehetoleg pfTrackben
- orientation
- track cleanUp
- solve cleanUp
- statistics

ELLENORZES

- geometria pfTrackben
- view lock
- maya nyilak-surface shader-es render es komp

3D INTEGRATION

- matchMove kezelese: group-ban, csak egyben, kulonben megtorlik a
matchMove

MAYA RENDER SETTINGS FOR TEST RENDER

- surface shader a proxy geometrián – kupo a lehető legtobb helyen,
De nem takarjak be egymast, jo lathato es lefedett legyen a legtobb hely
A footage-en es olyan helyekre is keruljanak kup, amit nem jelol locator,
Hogy ellenorizni lehessen, hogy megfelelo teret adott-e ki a matchmove!
- az imagePlane a renderCameran ki kell kapcsolni!
- maya software render command line-bol!
[renderFileName].bat segit!

Common tab

- | | |
|------------------------|------------------------------------|
| - file name prefix: | ld.: aktualis nevezektan |
| - frame animation/ext: | name.#.ext |
| - format: | .iff |
| - Frame padding: | 4 |
| - alpha chanell: | on |
| - width x height: | full resolution (pl.: 2048 x 1556) |

Software tab:

- | | |
|-----------------------|-----------------|
| - quality: | preview quality |
| - edge anti-aliasing: | highest quality |
| - shading/max: | 3/7 |

Motion blur eszo korben nem kell. – csak a rendert lassitja, a hatsokon nem latszik a kozel levo markereket pedig nagyon elmosza.

- motion blur: on

PFTRACK:

- gui
- shortcuts
- preferences
- 2d vs 3D view
- features - pl.: geo tracking, color correction, oFlow
mask, modeling, survey data, constrain,
feature tracking, color correction

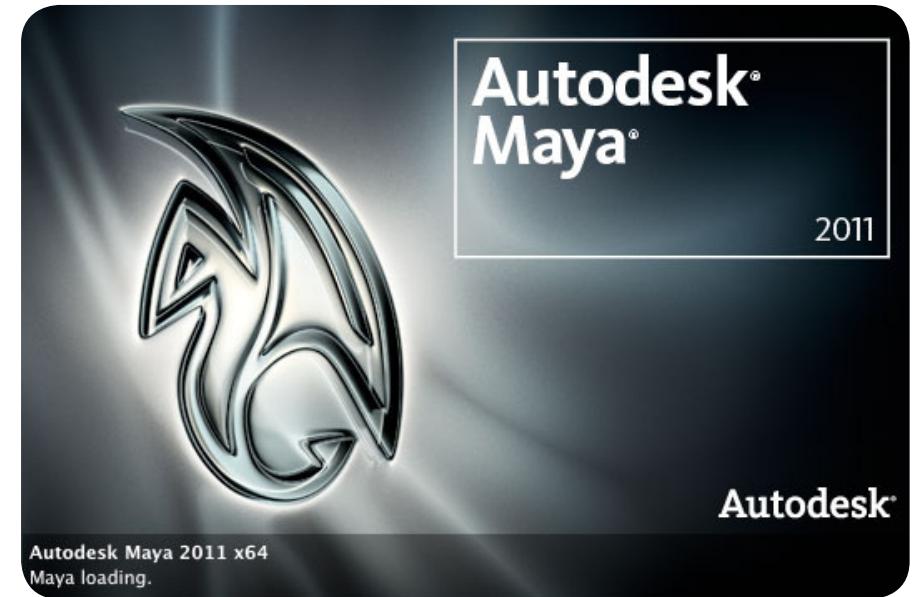
track ellenorzes: view lock

- project felepités

TIPPEK/TRUKKOK / MEGOLDASOK:

MAYA

GUI SETUP



Ez a rész abban fog segíteni, hogy mélyebben megismerd a Maya kezelőfelületét mintha csak menüzgetnél, és hagyján hogy sokkal gyorsabban melózz vele, megspórolj egy csomó felesleges, idegesítő, de leginkább unalmas menüzést és klikkelést. A mottó valami ilyesmi lehetne:

„Ha az `undo`-t, aminek a lehető legtöbb szoftverben,
szinte évtizedek óta ugyanaz a shortcut-ja még mindig menüből használod,
lehet el kéne mennen inkább erdésznek!”

Csak hogy a legfrissebb verziót használjam, a doksit Maya 2011-es verzióval irom 2010-ben, ez az ami jövőre jött ki az idén, de amennyire köztudottan fejlődik a maya, esélyes hogy nem lesz ezekkel bajod 2007-től 2026-ig.

A SCRIPT

A script az expression felett, de a programozás alatt van, de begyulladni azért nem kell, ha valamiben haverod a Maya akkor ez az. Semmi extrát nem csinalálsz majd scriptekkel, csak a fent emlitett klikkelgetés és menüzgetés helyett, a gyakran és ugyanúgy ismételgetett cuccokra fogod magad és csinálssz egy jó kis gombot. Ami miatt ez tényleg könnyű lesz és délutánra belejössz, az az a faja tulajdonsága a Maya-nak, miszerint – a legtöbb szoftverrel ellentében – fogja magát és egy ablakban akkor is kiirja mint script mindenzt amit a kezelőfelületen teszel ha nem akarod.

Ez az ablak a scripteditor, de erről kicsit később. Két nyelven is tudsz scriptelni Maya-ban. MEL-ben vagy Python-ban. A scripteditorban minden MEL-ben ír ki a maya, amivel nincs baj, emiatt fóleg ezt fogod használni, de ha már beletették, a python-t is érdemes tanulni, a legtöbb szoftver használja, mint általános programnyelv több dolgot is egyszerűbb benne megoldani, nem csak maya-s, hanem teljesen általános oldalakon is lehet hozzá segítséget találni, ráadásul szinte csak szintaktikai különböszég van a kettő között amihez pár nap alatt hozzá lehet szokni. Értsd: tök mindegy lesz, hogy a maya dokumentációban MEL-t vagy python-t nézel. Persze ha valamit simán úgy használssz ahogy a maya kidobja felesleges átirni, de ha már egy picit is szeretnél trükközni egyszerűbb lesz Pythonban.

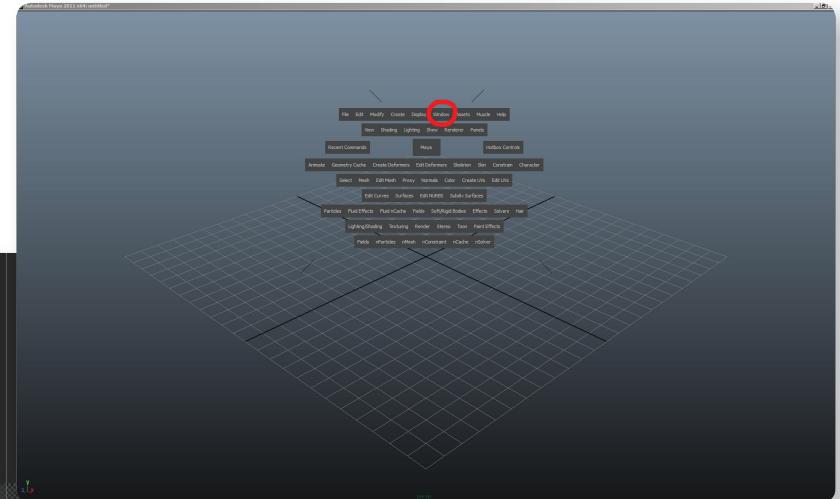
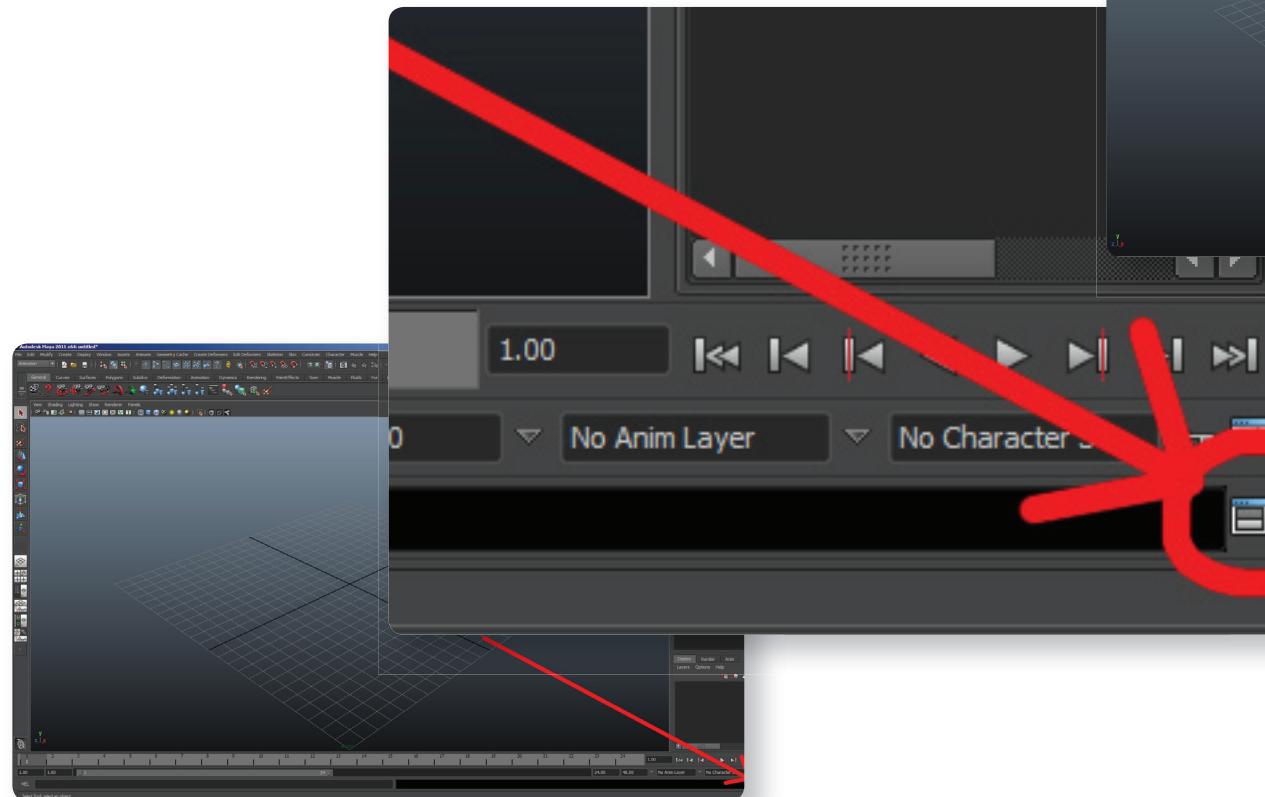
Ebben a doksiban megcsinálok egy két példát, de az egészről a lényeg az lesz, hogy pillanat alatt összedobj magadnak valami kis scriptet, legyen az bármi melótól teljesen független apró kis segítség és minél jobban ráérezz mikor érdemes és hogy állj neki. Szerencsére a maya-nak kitűnő a doksija ebben, mindenhez találssz majd részletes példát ami mayába bekopipésztelelve működni fog és ezekből nagyon egyszerűen ki lehet majd indulni, és akár bonyolultabb dolgokat is összeollózhatsz.

A maya doksit egyébként bookmarkold be az autodesk oldalrol: <http://download.autodesk.com/us/maya/2011help/index.html>

Alul a „Technikal documentation/commands” alatt találd ami kell, és tökmindegy hogy python vagy nem python, akármelyiket nézed, jobbra fent egy gombnyomással át tudsz váltani egyikről a másikra.

A SCRIPTEDITOR

A scripteditort több helyen éred el: window/general editors/script editor,
vagy:



és így néz ki és ezt tudja (nem sokat, nyugodtan használj különböző editort. pl.: notepad++) és ilyen amikor kiírja a mindent:

HISTORY

ez a maza helye

INPUT

ez a te helyed

The screenshot shows the Maya Script Editor window. At the top, there is a toolbar with various icons. Below the toolbar is a menu bar with 'File', 'Edit', 'Select', 'Transform', 'Display', 'Render', 'Shade', 'Light', 'Material', 'Effect', 'Utilities', 'Script Editor', 'Help', and 'Customize'. The main area is titled 'Script Editor' and contains a text editor with the following content:

```
// mental ray for Maya: successfully registered
// mental ray for Maya: loading startup file: C:/Program Files/Autodesk/Maya2011/mentalray/maya.rayrc
// parsing C:/Program Files/Autodesk/Maya2011/mentalray/include/AdskShaderSDKWrappers.mi
// generating Maya nodes...
// parsing C:/Program Files/Autodesk/Maya2011/mentalray/include/architectural.mi
// loading C:/Program Files/Autodesk/Maya2011/mentalray/lib/architectural.dll
// generating Maya nodes...
// parsing C:/Program Files/Autodesk/Maya2011/mentalray/include/base.mi
// loading C:/Program Files/Autodesk/Maya2011/mentalray/lib/base.dll
// generating Maya nodes...
// parsing C:/Program Files/Autodesk/Maya2011/mentalray/include/contour.mi
// loading C:/Program Files/Autodesk/Maya2011/mentalray/lib/contour.dll
// generating Maya nodes...
// parsing C:/Program Files/Autodesk/Maya2011/mentalray/include/paint.mi
// loading C:/Program Files/Autodesk/Maya2011/mentalray/lib/paint.dll
// generating Maya nodes...
// parsing C:/Program Files/Autodesk/Maya2011/mentalray/include/physics.mi
// loading C:/Program Files/Autodesk/Maya2011/mentalray/lib/physics.dll
// generating Maya nodes...
// parsing C:/Program Files/Autodesk/Maya2011/mentalray/include/production.mi
// loading C:/Program Files/Autodesk/Maya2011/mentalray/lib/production.dll
// generating Maya nodes...
// parsing C:/Program Files/Autodesk/Maya2011/mentalray/include/subsurface.mi
// loading C:/Program Files/Autodesk/Maya2011/mentalray/lib/subsurface.dll
// generating Maya nodes...
// parsing C:/Program Files/Autodesk/Maya2011/mentalray/include/surfaceSampler.mi
// loading C:/Program Files/Autodesk/Maya2011/mentalray/lib/surfaceSampler.dll
// generating Maya nodes...
updateRendererUI;
updateRendererUI;
saveShelf General "C:/Users/ford/Documents/maya/2011-x64/prefs/shelves/shelf_General";
// 1 //
// Saving preferences to : C:/Users/ford/Documents/maya/2011-x64/prefs/userPrefs.mel
// Saving window positions to : C:/Users/ford/Documents/maya/2011-x64/prefs/windowPrefs.mel
// Saving runtime commands to : C:/Users/ford/Documents/maya/2011-x64/prefs/userRunTimeCommands.mel
// Saving named commands to : C:/Users/ford/Documents/maya/2011-x64/prefs/userNamedCommands.mel
```

Below the text editor, there are tabs for 'MEL' and 'Python'. The bottom of the window features a numeric keypad from 1 to 9 and a status bar with the text 'pu sp'.

Ha ez megvan örülünk, vágjunk is bele!

„HELLO WORLD” HELYETT...

Mint matchmover, kamerákkal esélyes hogy majd babrálsz, szóval kezdjük olyasmivel, hogy lerakok egy kamerát és beállítok minden. Pl.: focal length, film back, satöbbi, csak úgy simán, egérrel-kézzel.

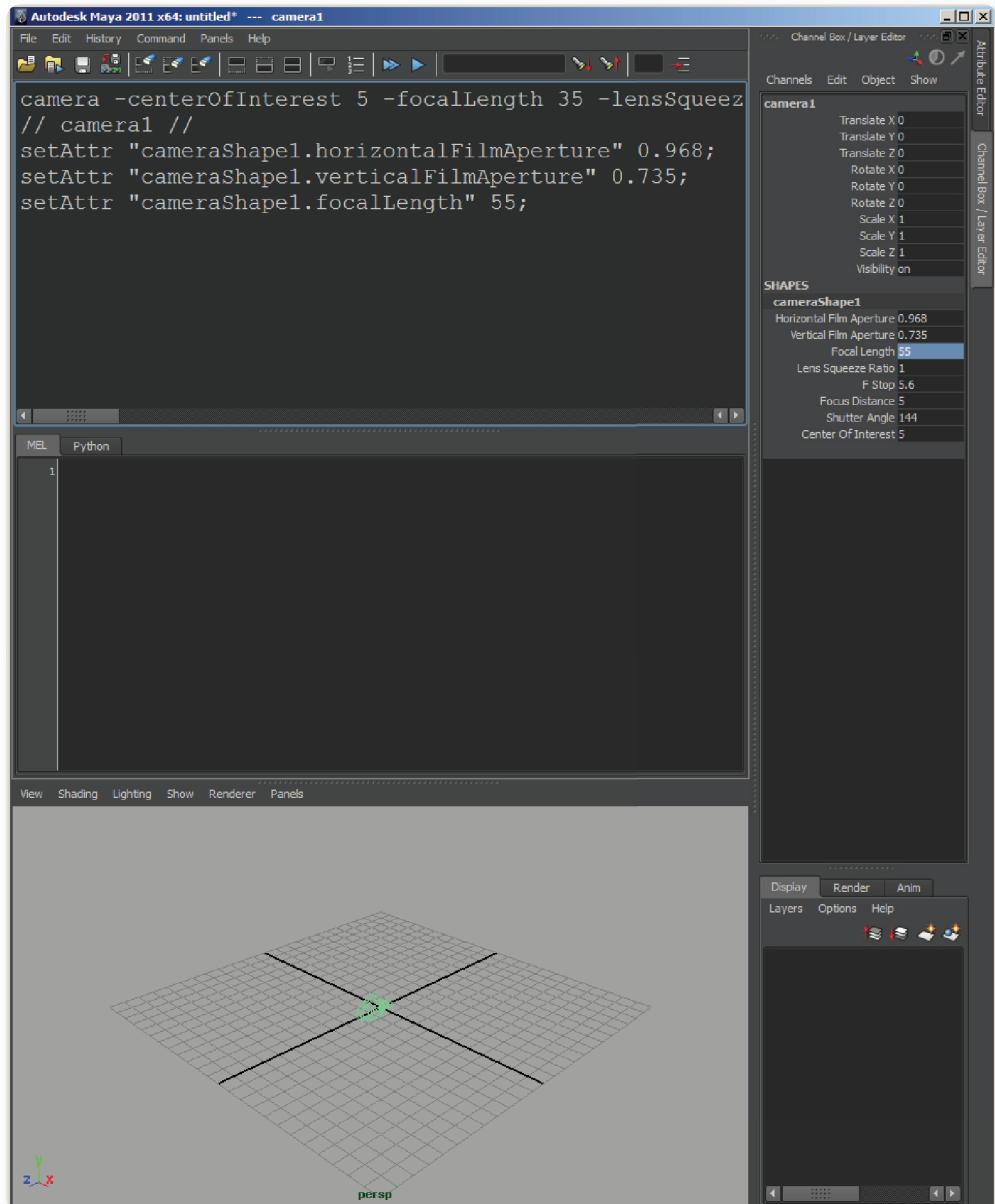
Itt tratunk:

camera megjelent, parameterek beállítva, scripteditor felül, script kiirva, pont ahogy csináltam create menüben egy kamerát és beirtam az értékeket channelbox-ba.

Innentől elég sokáig el lehet lenni azzal, hogy kikopipészteled a histroybol ami kell, beirod alulra, kijelölök és középső egérgombal felhúzod shelfre.

Ennél a pontnál két dologra kell csak figyelni. Az egyik, hogy az input ablakban külön tab van pythonnak és melnek. Keverni nem lehet a kettőt, vagy python vagy mel scriptet futtatsz és tabonként külömbözteti meg a maya. Ha rossz tabba irsz, hiába jó a script hibát kapsz.

A másik, hogy a maya a lefutott scriptek eredményeit, visszajelzéseket is kiir, amiket ha scriptben hagysz gondot ugyan nem nagyon okoznak, de össze vissza fogják kavarni a scriptet, pláne többsoros visszajelzések esetén.



és akkor ugyanez scriptként:

Amit pedig itt jó tudni az annyi, hogy control enter-rel csak a kijelölt rész fut le a scriptből és inputban hagyja a maya.

Az inputban hagyja azért jó, mert máskülömben nem, és eltünik a scripted, a csak a kijelölt rész azért jó, hogy ha hiba van soronként végig tudd ellenőrizni hol a hiba.

