

FÉNYSGUGÁR-REKONSTRUKCIÓ TETSZŐLEGES NÉZŐPONT KIALAKÍTÁSÁHOZ

RAY RECONSTRUCTION TO MAKE ARBITRARY VIEWPOINT

Illés Attila ^{1*}, Koszna Ferenc ¹, Kátai-Urbán Gábor ¹, Megyesi Zoltán ¹

¹ Informatika Tanszék, GAMF Műszaki és Informatikai Kar, Pallasz Athéné Egyetem, Magyarország

Kulcsszavak:

Free-Viewpoint Video, rekonstrukció, tetszőleges nézőpont

Keywords:

Free-Viewpoint Video, reconstruction, arbitrary viewpoint

Cikktörténet:

Beérkezett 2016. szeptember 20.
Átdolgozva 2015. november 24.
Elfogadva 2015. november 25.

Összefoglalás

Ebben a cikkben bemutatunk egy eljárást, amellyel egy kalibrált kamera rendszerben a kamerák által megfigyelt színtérről tetszőleges nézőpontból készíthetünk nézeti képet. A cikk bemutat egy szintetikus képi-adathalmaz készítésére alkalmas módszert is. Ezzel az adathalmazzal fogjuk tesztelni a tetszőleges nézőpont kialakítására készített eljárásunkat.

Abstract

In this article we introduce a method to make an arbitrary viewpoint in a calibrated camera system. We made a synthetic image-set with a tool which will be introduced as well. For testing of our tool which make an arbitrary viewpoint we used image-set what is made by our tool formerly.

1. Bevezetés

Napjainkban egyre inkább jellemző az interaktivitás megjelenése a szórakoztató iparban. A DVD lemezek megjelenésével már lehetőség volt arra, hogy több előre definiált nézőpontból tekinthessünk meg egy-egy jelenetet. Bizonyos 3DTV-k fejlődésével a tetszőleges nézőpont kialakítása is lényeges szempont lett. A tetszőleges nézőpont a megfigyelt színtérről olyan pozíciókból is enged betekintést, amelyekre részlegesen vagy egyáltalában nem lát egy kamera sem. Ezekben a nézőpontokban a kívánt nézeti képet olyan képekből kell megalkotni, amelyeken az adott színtér látható más-más nézetekből. Előfordulhat, hogy a kívánt nézőpontból olyan térrész lesz látható, amelyet nem minden kamera lát be teljesen, így sok esetben hiányos információkból kell dolgoznunk. Ezt a területet hívják Image-Based Rendering-nek[1][2], vagy Free-Viewpoint Video-nak [3], azaz képalapú szintézisnek, ahol a színtérről nem készül geometriai modell, és így a színtér objektumait sem definiáljuk geometriai tulajdonságokkal, mint pontthalmaz vagy egyéb modell reprezentáció.

Ebben a cikkben bemutatunk egy eljárást, amely képes egy mesterséges színtérről tetszőleges nézetet generálni függetlenül attól, hogy a színtéren mi található. A fényugár-rekonstrukció azt jelenti, hogy a kívánt tetszőleges nézeti kép nem minden pixeléhez tartozik olyan fényugár, amelynek színét ismerjük. Ezért a fényugár-rekonstrukció során megszínezzük ezeket a fényugarakat, amelyek a nézeti kép hiányzó pixeleit is megszínezi.

A módszer tesztelését szintetikus képi adathalmazzal valósítottunk meg, melynek előállítását is bemutatjuk a cikkben. A legvégén kiválasztottunk egy módot, amellyel összehasonlítottunk a referencia és a szintetizált képek között. A tesztek után elmondhatjuk, hogy ez az eljárás képes szintetizálni egy mesterséges színtér tetszőleges nézeti képeit. Itt fontos megjegyezni, hogy

* Kapcsolattartó szerző. Tel.: +36 76 516 447; fax: +36 76 516-399
E-mail cím: illes.attila@gamf.kefo.hu

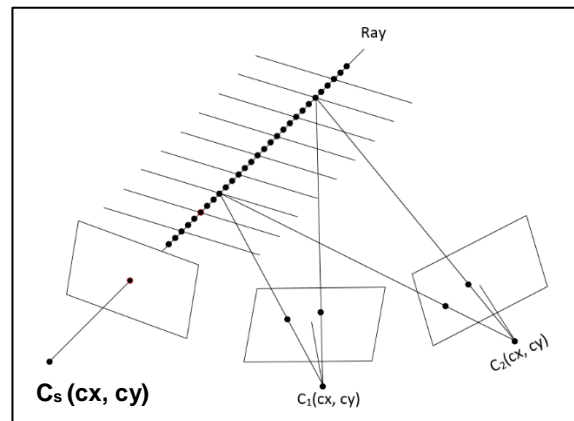
bizonyos megszorításokat figyelembe kell venni az eredmény kiértékelésekor. További munka, hogy összetett mesterséges színtereken, illetve valós színtereken is teszteljük a módszerünket.

2. Fénysugár-rekonstrukció

A cél az eljárásunkban az, hogy az általunk kívánt tetszőleges nézőpontban lévő nézeti képet megalkossuk. Kalibrált kamera rendszert használva, definiálhatunk n darab valódi kamerát (C_1-C_n) és a tetszőleges nézőpontokhoz egy-egy szintetikus kamerát ($C_{s1}-C_{sm}$). Esetünkben több valódi kamerát és egy darab szintetikus kamerát definiáltunk. A valódi kamerák szolgáltatják a bemeneti képeket és ezek felhasználásával megalkothatjuk a szintetikus kameránk által megfigyelt térrész nézeti képét. Az 1. ábrán látható az elv, amivel előállítjuk a szintetikus nézeti képet. A szintetizált fénysugarak nem részei a valódi kamerák által definiált fénysugármezőnek. A C_s kamera készíti (szintetizálja) a kívánt nézeti képet. Az eljárás lényege, hogy minden egyes képpontot a szintetikus képen meg kell színezni valamilyen színnel. Ahhoz, hogy eldöntsük, hogy milyen színnel színezzünk, szükség van a fénysugár-rekonstrukciós eljárásra [1].

Az eljárás menete az, hogy minden egyes képponton keresztül lövünk egy sugarat (Ray), amely a szintetikus kamera középpontjából (cx, cy) indul. Ezekon a sugarakon végighaladunk Z -ben távolodva a kamera középponttól. A sugáron való haladás léptéke azt definiálja, hogy a színteret milyen Z értékeknél szeretnénk egy-egy X - Y síkkal elmetszeni.

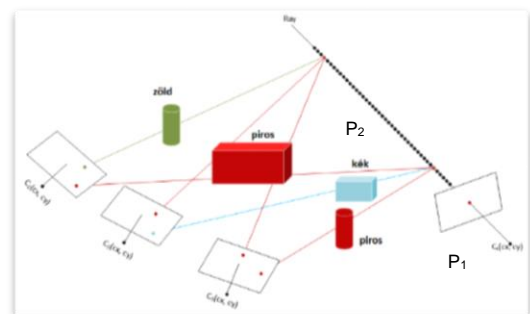
Végighaladva ezen a sugáron a léptéknek megfelelően minden egyes 3d pontot a sugáron levetítünk [5] a bemeneti kamerákra. Így a bemeneti képeken megjelölünk egy-egy képpontot, amelyeknek színét visszaadva keletkezik egy halmaz. Ez a halmaz tartalmazza, hogy a sugáron az adott 3d-s pont milyen színűnek látszik az egyes bemeneti kamerák képein. Ezekből a színértékekből kell kiválasztanunk egyet, melyet felhasználva meghatározhatjuk, hogy milyen színnel színezzhetjük meg a szintetikus képünk adott pixelét. Azt, hogy az adott lépésekben, a keletkezett színértékek halmazaiból melyik színt használjuk fel, azt a mi módszerünk fogja meghatározni. Meg kell határozni a sugáron azt a konzisztens pontot, amelyhez az a konzisztens szín tartozik, amellyel meg kell színezni a szintetikus kameránkhoz tartozó kép adott pixelét. Ezek nem triviális problémák.



1. ábra: Szintetikus és valódi kamerák

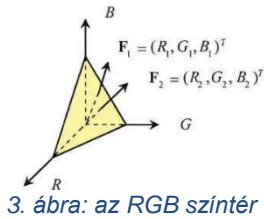
3. Konzisztens pozíció és szín meghatározása

A konzisztens pont meghatározása nem triviális probléma. A 2. ábrán látható szemléltetésben is jól látszik, hogy a sugár adott pontjaihoz rendelhető színértékek meghatározása nem egyértelmű feladat. Feltesszük, hogy a Lambert modell [6] itt is érvényes, tehát egy felületről visszaverődő fénysugár minden egyes kamerán ugyanolyan színűnek látszik. A szemléltetéshez a színtéren egy zöld és egy piros henger, egy piros téglatest és egy kék téglatest van elhelyezve. A C_1-C_n kamerák képei alapján szeretnénk meghatározni egy-egy 3d-s ponthoz tartozó színértéket a sugáron. A 2. ábrán két 3d-s ponthoz mutatjuk be a lehetséges színértékeket. A P_1 pont pirosnak látszik a C_1 és a C_3 -as kamera képein és kéknek a C_2 -es kamera képein. A P_2 -es ponthoz tartozó színértékeket a C_1 -es, C_2 -es és a C_3 -as kamera képe is befolyásolja. A C_1 -es kamera képein zöldnek látszik, a C_2 -es és C_3 -as kamerán pedig pirosnak. Ebből jól látszik, hogy egy-egy sugárponthoz tartozó színérték meghatározása nehéz feladat. Ebben a cikkben egy lehetséges megoldást mutatunk be, amely a sugárpontokhoz való színérték rendelést egy egyszerű



2. ábra: Egy szintetizált pixel

átlagszámítással old meg. Ha levetítettük a sugár adott 3d-s pontját minden egyes bemeneti kamera képére, akkor keletkezik n darab színérték. Ezeken elvégezve egy átlagszámítást megkapjuk az átlagszín értékét. A további eljárások teszteléséhez ez most elegendő. A későbbiekben szükség lehet, ennek a színérték-kiválasztásnak az optimalizálására is.



Az átlagolást elvégezzük minden egyes sugárponton az adott lépték és intervallum szerint, ahol a sugarat bejárjuk. Így keletkezik annyi színérték, mint ahány lényeges pontot bejártunk a sugáron. Ezeken a színértékeken elvégzünk egy szórás számítását, segítségül hívjuk az RGB színtérben definiálható távolságokat. 3. ábrán látható az RGB színtér vektoriális értelemben. A (1) képlet mutatja be egy RGB érték távolságának kiszámítását az origótól.

$$d_{RGB} = \sqrt{R^2 + G^2 + B^2} \quad (1)$$

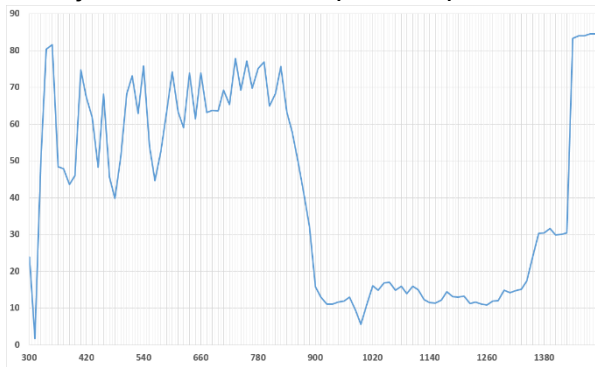
Ha kiszámoltuk az összes lényeges színérték távolságértékét (x_i) az f képen, akkor kiszámítjuk az átlagukat, azaz az átlagos távolságértéket (\bar{x}). Ezt az átlagot fogjuk felhasználni a szórásértékek meghatározásához. A szórásértékek közül a legkisebb jelöli majd számunkra a konzisztens pontot. A (2) képlet a szórásnégyzet számítását végzi, a gyökvonás után a szórásértéket kapjuk meg.

$$Var(x) = \sigma^2 = \frac{\sum_{i=1}^k f_i (x_i - \bar{x})^2}{\sum_{i=1}^k f_i}, \quad \sigma = \sqrt{\frac{\sum_{i=1}^k f_i (x_i - \bar{x})^2}{\sum_{i=1}^k f_i}} \quad (2)$$

Ezen értékek meghatározása után felvehetünk egy sugár mentén a színértékekhez tartozó szórásértékek grafikonját. Két esetet vizsgáltunk meg, amikor a konzisztens pontot elég jól megtaláltuk, és amikor hibás pontot választott az eljárás, ilyenkor kívánt nézeti képen az aktuális pixel színe nagyban eltér attól, amit elvártunk volna. A 4. ábrán látható egy olyan grafikon, amin az látható, hogy az eljárás a megfelelő pontot választja ki könnyedén. A kiválasztást jelenleg úgy oldottuk meg, hogy a sugár mentén összegyűjtött szórásadatokon egy minimum keresést végzünk. A kisebb érték képviseli a jobb eredményt. Ebben az esetben a minimum keresés megtalálta azt a pozíciót, amely megfelelő számunkra, hogy megszínezzük vele fénysugarunkat, majd ezzel a kimeneti kép adott pixelét is. Az 5. ábrán olyan esetet mutatunk be, ahol a minimum olyan helyen van, ahol nem a megfelelő eredményt adja. Mindezt onnan tudhatjuk, hogy ismerjük a tesztrendszerünk bemenő paramétereit, illetve ismerjük a referencia kép adott pixelének színét. Feltételezzük, hogy egy simítás az ábrázolt



4. ábra: Szórásértékek egy jó esetben



5. ábra: Szórásértékek rossz esetben

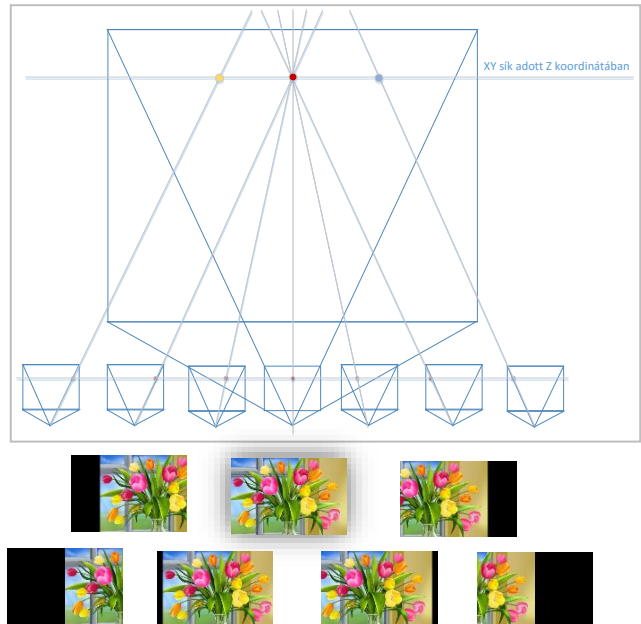
pontokon a megfelelő eredmény hozná ebben a pár vizsgált esetben. A konzisztens pont meghatározását valós esetekben is kell vizsgálni, hiszen a bemutatott eljárás csak néhány paramétert figyel a lehetséges paraméterekből. A konzisztens pont kiválasztása más módszerekkel további paraméterek bevonását teszi szükségessé. Ilyen paraméterek lehetnek a Z-érték figyelembe vétele, a kamerák egymáshoz képesti elhelyezkedése, illetve a fénysugarak beesési szöge a vizsgált sugárpontba. Ezek mellett számos paramétert lehet bevonni a vizsgálódásba.

4. Tesztkörnyezet

4.1. Szintetikus képi adathalmaz előállítása

Ahhoz, hogy az módszerünket tesztelni tudjuk, szükség volt egy tesztkörnyezet elkészítésére. Létrehoztunk egy szintetikus képi adathalmaz előállító eszközt. Itt a kalibrált kamerarendszer egy bemeneti és n darab kimeneti kamerával rendelkezik. Az egyszerűség kedvéért a kamerákat egy síkban helyeztük el, és csak az x tengely mentén toltuk el 300-al pozitív és negatív irányban is. A Teszthez 10-10 kamerát helyeztünk el pozitív és negatív irányban egyaránt. A bemeneti kamerát az origóban helyeztük el.

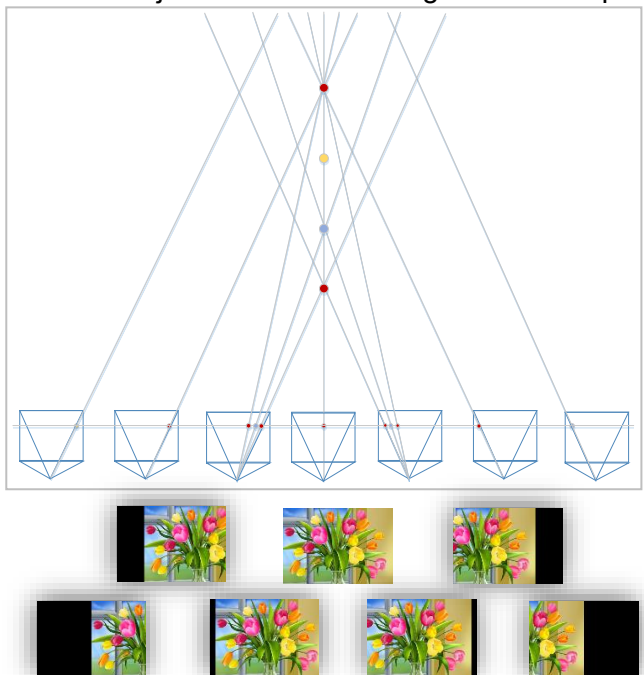
A generálás során, a kimeneti képek minden egyes pixelén végigmegyünk egyesével. A kimeneti kamera középpontjából egy sugarat lövünk ki az aktuális pixelen át, s az általunk megadott vágási mélységben ezt a sugarat elmetsszük. A metszéspontot levetítjük a bemeneti kamerára, és a bemeneti képen az adott pixel színével megszínezzük az adott kimeneti kép aktuális pixelét. Ezt végig futatjuk az összes kimeneti kamera összes pixelén. Az eredmény a 6. ábrán, az alsó részen látható. Ez a konfiguráció azt eredményezi, hogy a bemeneti képet arra a Z értékre vetítjük ki, amelynél a fénysugarakat elmetsszük a kimeneti képeken. Másképp fogalmazva olyan, mint egy moziban ülve. A bemeneti kép a vetítógép, ahol a vászon adott Z -ben van és a kimeneti képek azok lesznek, amiket a közönség lát egy-egy székéből nézve, feltételezve, hogy mindenki előre néz.



6. ábra: Szintetikus képek előállítása

4.2. Tetszőleges nézőponti kép előállítása

A cikk lényeges része az a módszer, amivel előállíthatjuk a kívánt tetszőleges nézeti képet több valódi kamera képi adathalmazára támaszkodva. A módszer tesztelésére az előre legyártott szintetikus képi adathalmazból, mint bemeneti képek halmaza, fogunk dolgozni. Ebben az esetben a kimeneti kamerából lesz egy darab és a bemeneti kamerából n darab. A kimeneti kamera a tetszőleges nézeti képet állítja elő 7. ábra. Most a kimeneti kép pixelein haladunk egyesével végig és a kilövünk egy-egy sugarat a kamera középpontjából az adott pixelen át. Ezen a sugáron végighaladva a beállított léptéknek megfelelően minden egyes lényeges 3d-s pontot a sugáron levetítünk a bemeneti kamerákra. Ezek után a konzisztens pontot és a konzisztens színt kell meghatározni, a 3. pontban ismertetett módon, ezután megszínezzük a kívánt nézeti kép pixeleit. Az aktuális fénysugáron való végighaladás paraméterezhető, így a tesztelés során különböző intervallumokat és különféle léptéket tudunk alkalmazni és a tesztelés során



7. ábra: A tetszőleges nézet előállítása

felhasználni. Onnan tudhatjuk biztosan, hogy jó-e az eredmény, hogy ugyanazt a képet kell kapnunk, mint a referencia kép.

5. Eredmények

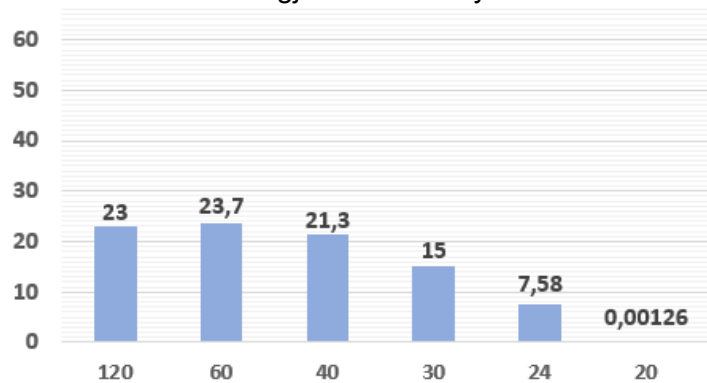
A módszerünket tesztelve arra a következtetésre jutottunk, hogy a szintetikus képi adathalmazból jól dolgozott, bár volt olyan eset, ahol nem hozta a várt eredményt. Kellott egy módszer, amely szintetizált kimeneti kép minőségét ellenőrzi. Rendelkezünk a kiindulási képpel, amelyből a szintetikus nézeti képek készültek. Ez lesz a referencia kép, amelyet a kívánt tetszőleges nézeti kép előállítás után egymással összehasonlítunk. Egy egyszerű összehasonlítást végeztünk. A 3. képlet felhasználásával, amely különbséget vizsgál két kép között. A képek megjelölése a következő: f_r a referencia és f_e az eredmény kép.

$$diff_{RGB} = \sum_{i \in N} d_{RGB}(f_r(i), f_e(i)) \quad (3)$$

A különbséget az azonos pozíciójú pixelek színértékeire vizsgáljuk. Hiszen a referencia képet kell visszakapnunk. Végeztünk vizsgálatokat különböző léptékek esetén is. Az 1. táblázat tartalmazza a vizsgált paramétereket, amelyek alapján a vizsgálatokat végeztük. A 8. ábrán megjelenítettük a jósági tényezőket adott lépték esetén. Ebben a konfigurációban minden fénysugárnál csak a 300 és 1500 közötti Z értéket vettünk figyelembe. Az osztás azt jelenti, hogy az előbbi intervallumot hány részre osztjuk. A lépték Z-ben megadja, hogy a fénysugáron mely pontokat fogjuk levetíteni a bemeneti kamerák képeire. Tekintsünk meg egy példát. Vegyük az első oszlopot. Az osztás 10, így a 300 és 1500 között intervallumot 10 részre osztjuk. Ez $(1500-300)/10 = 120$ léptékre jön ki. Ekkor a lényeges pontok a fénysugáron 420, 540, ..., 1380, 1500. A jósági tényezőt a 4. képlet felhasználásával határoztuk meg.

$$diff(x, y) = \frac{\sqrt{\sum_{i=1}^n |x_i - y_i|^2}}{S_{Count}} \quad (4)$$

A jósági tényező egy hibára utaló szám. Minél kisebb annál jobban hasonlít a referencia képre. Az 1. táblázatban a 60-as osztás hozta a legjobb eredményt.



8. ábra: A jósági tényezők

1. táblázat: Jósági tényezők vizsgálatának eredményei

Osztás	10	20	30	40	50	60
Lépték Z-ben	120	60	40	30	24	20
Jósági tényező	23	23,7	21,3	15	7,58	0,00126

Az eredményeket képeken is bemutatva látható a minőségi javulás. A 9. kép a referencia képet mutatja, amiből a szintetikus képeket készítettük a bemeneti képi adathalmazhoz. A 10. kép a referencia képből kivágott részlet, amit összehasonlíthatunk a tetszőleges nézőpontban szintetizált képeinkből kivágott részletekkel. A kis részletek balról jobbra haladva a következők (referencia, 10, 30,60). A számok az osztást jelöli, amivel a tesztek paraméterezve lettek.



9. ábra: Referencia kép



10. ábra: kivágott részletek

Érdeemes megfigyelni, hogy a 10-es osztásnál milyen sok a hibás színű pixel. A 20-as osztásnál nem sok javulás figyelhető meg, míg a 60-as kimagaslóan jó eredményt mutatott. Szinte tökéletes fénysugár-rekonstrukciót hajtott végre. Azt, hogy a lépték növelésével miért romlik vissza a minőség, azzal lehet magyarázni, hogy a megfelelő konzisztens pont kiválasztásakor sokkal több adatból kell meghatározni a jó értéket. Ez jelzi felénk, hogy az eljárásunkat finomítani kell és a konzisztens pont meghatározására más paramétereket is figyelembe kell venni. Hiszen minél több adatból szeretnénk meghatározni a konzisztens pontot, s az ahhoz tartozó színértéket, annál több olyan színérték lehet a halmazban a szórás számításánál, ami nem a megfelelő szín.

6. Konklúzió és továbbfejlesztési lehetőségek

Az eredmények azt mutatják, hogy az eljárás, amit kidolgoztunk sok esetben ad jó megoldást. A módszert tovább kell finomítani, több paramétert kell bevenni a konzisztens pontok meghatározásához. Fontos megjegyezni, hogy a színtér most két dimenziós képet tartalmazott. A továbbiakban 3D-s objektumokat kell elhelyezni a színtérre, s azon elvégezni a tetszőleges kameranézet előállítását. A takarási probléma ott újabb problémákat fog felvetni, amelynek a megoldására az eljárásunkat is tökéletesebbé kell tennünk. Érdeemes megtekinteni Andrew Fitzgibbon cikkét [4], amely más megközelítésben választja ki azt a színértéket, amellyel megszínezi a fénysugarakat. A továbbiakban e két módszer összehasonlítása is érdekes lehet a számunkra.

Köszönetnyilvánítás

Köszönet a Pallasz Athéné Egyetemnek, hogy lehetőséget biztosítanak a doktori tanulmányok végzéséhez.

Irodalomjegyzék

- [1] S. B. Kang. A survey of image-based rendering techniques. In *VideoMetrics, SPIE Vol. 3641*, pages 2–16, 1999.
- [2] Heung-Yeung Shun, Sing Bing Kang, Shing-Chow Chan, "Survey of Image-Based Representations and Compression Techniques," *IEEE Trans., Circuits Syst., Video Technol.*, vol.13, no. 11, pp. 1020-1037, 2003.
- [3] Bastian Goldlücke, *Multi-Camera Reconstruction and Rendering for Free-Viewpoint Video*, Ph.D. Thesis, 2006.
- [4] Andrew Fitzgibbon: *Image-Based Rendering Using Image-Based Priors*, *International Journal of Computer Vision* 63(2), 141–151, 2005
- [5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [6] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007. p.93.