

Prototip de interacțiune în medii de realitate îmbunătățită bazată pe recunoașterea traseelor 2D ale marcajului

Rareș Bradi

Universitatea Tehnică Cluj-Napoca

Str. Memorandumului nr. 28, Cluj-Napoca

bradi.rares@gmail.com

REZUMAT

În acest articol se prezintă un prototip de interacțiune pentru aplicațiile de realitate virtuală îmbunătățită bazată pe recunoașterea căii pe care marcajul urmărit o parcurge de-a lungul unui interval de timp. Ea a fost realizată utilizând ARToolkit pentru urmărire și algoritmul *one-dollar* pentru recunoașterea traseului. Traseul străbătut de marker este stocat, normalizat, rotit și scalat și în final comparat la unghiuri diferite cu un set de șabloane pre-înregistrate. La sfârșit, după comparare va rezulta un scor de similitudine. Scorul rezultat este transformat într-o măsură de încredere și comparat cu un prag pentru a da o decizie finală de potrivire sau respingere. O analiză asupra soluției este desfășurată de-a lungul unui scenariu care scoate în evidență eroarea și calitatea interacțiunii.

Cuvinte cheie

Realitate virtuală îmbunătățită, potrivire șabloane, tehnică de interacțiune.

Clasificare ACM

H5: Information interfaces and presentation (e.g., HCI), H.5.2: User Interfaces, H.5.m: Miscellaneous; B.4: Input/output and data communications, B.4.2: Input/output devices; D.1.5: Object-oriented programming

INTRODUCERE

Realitatea virtuală îmbunătățită (*Augmented Reality*) - AR este o disciplină extinsă din realitatea virtuală care are ca scop suprapunerea informațiilor, modelelor grafice peste scene din lumea reală. În realitatea virtuală, utilizatorul este cufundat într-un mediu sintetic în care nu are contact cu lumea exterioară. Spre deosebire de acesta, în realitatea virtuală îmbunătățită, vizualizarea lumii reale se face împreună cu obiecte virtuale care sunt fie compuse, fie suprapuse cu acesta din urmă. Astfel această disciplină augmentează sau îmbunătățește percepția noastră asupra lumii.

Azuma și Baillot a enumerat în [1] principalele caracteristici ale sistemelor AR: 1. Combinarea spațiului real cu spațiul virtual; 2. Poziționare tri-dimensională: informația grafică este aliniată spațial corect față de împrejurimile utilizatorului. Pentru a obține o înregistrare corectă, mijlocul prin care captăm scena reală trebuie să poată fi urmărit; 3. Interactivitate în timp real: elementele prezentate utilizatorului sunt generate în timp real. Trasarea acestora depinde de intrările utilizatorului și schimbărilor survenite mediului. Ultima caracteristică a fost abordată în diferite moduri în literatură.

Multe cercetări făcute în AR erau în domeniul algoritmilor de poziționare (*registering*) și urmărire (*tracking*) video ale unor trasături ale scenei. Progresul a permis migrarea aplicațiilor AR tip desktop pe platforme mobile. Algoritmii din urmă acoperă însă doar prima și a doua caracteristică din cele trei enunțate. Atingerea interactivității în astfel de medii este dificilă întrucât: 1. nu avem un dispozitiv de intrare standard ale cărui date să fie certe sau disponibile în orice moment; 2. mediul este perturbat de diferite erori (poziție, rotație) date de algoritmi de mai sus; 3. apar condiții specifice (ocluziune) ce pot incapacita pe termen scurt sau dereglă un astfel de sistem; 4. adaptarea conținutului suprapus la mediu real prin extragerea condițiilor de mediu (luminozitate)

Din această cauză tehnicile de interacțiune existente nu pot fi aplicate direct. Adaptarea lor necesită cunoștințe despre tehnica originală (model) împreună cu cele despre contextul de folosire.

Lucrarea de față propune un prototip prin care se pot realiza tehnici de interacțiune combinând aceste cunoștințe. Procesul de poziționare și urmărire se face prin ARToolkit [11]. Modelul îl reprezintă gestul, care în medii ce dispun de dispozitive clasice de intrare puteau fi identificate prin algoritmi de recunoaștere a formelor (*pattern matching*). Deoarece trăsăturile de bază (poziție, rotație) folosite de aceștia sunt perturbate de erori în sisteme AR, s-a propus restricționarea recunoașterii formelor doar la cele 2D și alegerea unui algoritm care să folosească doar trăsăturile cele mai puțin afectate de erori (poziția). Algoritmul propus de Wilson et al. [10] este invariant la rotații, scalări sau translații și face o potrivire a unor puncte 2D ordonate ce alcătuiesc forma de recunoscut (formă candidat) cu un set pre-stocat de puncte 2D numit formă referință sau șablon. În urma potrivirii unui candidat cu fiecare șablon pre-stocat va rezulta un scor de potrivire. Acestea sunt combinate pentru a da o măsură de încredere dacă un candidat se regăsește în setul de șabloane. Aceasta ajută la diminuarea erorii. Din cauza condițiilor de mai sus este nevoie de o logică care să extragă informațiile utile din context (ARToolkit) și să interfațeze cu modelul (captarea punctelor și asamblarea lor într-o formă). Logica s-a implementat printr-un automat de stări finite care să trateze primele trei dificultăți de mai sus. Evaluarea soluției s-a făcut printr-un caz de utilizare de navigare printr-un meniu virtual. Scopul a fost determinarea ratei de erori la recunoaștere și a calității interacțiunii la utilizator. În continuare lucrarea este structurată după cum urmează. În secțiunea REALIZĂRI ASEMĂNĂTOARE sunt prezentate

prototipuri de interacțiune diferite și care exploatează aspecte diferite ale mediului AR pentru realizarea interacțiunii cu utilizatorul. Secțiunea DESCRIEREA PROTOTIPULUI oferă detaliile legate de structura și implementarea lui. În EXPERIMENTE se descriu condițiile de testare, măsurile folosite și configurațiile componentelor din secțiunea anterioară. REZULTATE ȘI CONCLUZII raportează performanțele obținute și încheie prin observațiile găsite.

REALIZĂRI ASEMĂNĂTOARE

Piekarski descrie în [6] o modalitate prin care se poate urmări poziția mâinii în coordonate 3D folosindu-se de ARToolkit. Matricea de transformare returnată în mod uzual de bibliotecă este atașată unui model distorsionat al camerei a cărei acuratețe depinde de modelul de calibrare folosit. Pentru contracararea acestei insuficiențe se aplică o transformare ortogonală asupra unor parametri intrinseci din fișierul de configurare a camerei. Metode de interacțiune folosind doar aceste informații sunt însă greu de realizat din cauza erorilor ce pot surveni în determinarea rotației precum și a adâncimii. Mason și Billingham prezintă în [5] o modalitate de realizare a unui dispozitiv mouse într-un mediu AR. Soluția adoptată împotriva problemei mai sus menționată a fost de a normaliza spațiul 3D în care este urmărit dispozitivul. Impunând anumite constrângeri volumului de vizualizare, anumiți parametri (ex rotația pe axa Z) se pot determina cu o precizie ridicată. Aceștia pot fi asignați diferitelor operații specifice aplicației. Diferitele scenarii implementate (joystick, mouse cu buton pentru derulare) demonstrează posibilitatea atingerii unui control fin în mediu AR fără a pierde din naturațea interacțiunii sau din timpul de răspuns la aceasta. Abordarea luată în lucrarea de față diferă de cele descrise până acum, prin faptul că se limitează la gesturile în domeniul 2D, făcând uz doar de coordonatele ecran. Deși prin aceasta, gradul de libertate al gesturilor de interacțiune sunt mai limitate, rata de eroare este mai scăzută deoarece ARToolkit are o acuratețe ridicată pentru urmărirea poziției marcajului. O altă motivație este faptul că în unele clase de aplicații precum cele pentru platforme mobile tehnica cea mai comună de interacțiune sunt căile 2D.

În cele mai dese cazuri conținutul suprapus este de natură vizuală dar definiția enunțată de Azuma și Baillet în [1] nu limitează acest aspect. Astfel avem lucrări precum [3] în care se folosesc marcasele acoperite de imagini, pentru a ajuta identificarea și aranjarea surselor sonore în scenă. Percepția realității este îmbunătățită prin adăugarea sunetului 3D. Ansamblul de cost redus format din cameră web și marcasele de cadru permit poziționarea de surse sonore în spațiul 3D. Prin integrarea bibliotecii de sunet EAX de la Creative în ARToolkit i s-a putut defini fiecărui element sonor în scenă proprietăți specifice. Obiectele sonore puteau fi poziționate prin intermediul unui indicator echipat cu marcaj. Utilizatorul putea să-și formeze o impresie bună despre efectul generat de surse dintr-un astfel de plasament. Principala limitare este necesarul existenței unui marcaj mare în fundalul imaginii camerei peste care erau suprapuse toate obiectele sonore. Dezavantajul acestuia se observă la apariția ocluziunilor,

de exemplu momentele în care marcajul de poziționare acoperea pe cel folosit pentru generarea scenei.

Dacă ne referim doar în contextul de recunoaștere a gesturilor mâinilor în aplicații AR atunci unul dintre cele mai recente sisteme este cel descris de Radkowski și Stritzke în [8]. Tehnicile de interacțiune puse la dispoziție de sistem facilitează manipulearea de părți ale unor obiecte virtuale în vederea asamblării lor. Componenta hardware folosită constă dintr-o cameră web și o cameră Kinect. Prima cameră este folosită de ARToolkit pentru captarea scenei și pentru poziționarea bucăților virtuale în aceasta. În acest scop se folosesc marcase. Camera Kinect este folosită pentru captarea gesturilor mâinilor. Informațiile de adâncime și cadrele captate de ea sunt trimise unui sistem de recunoaștere a mâinilor, bazat pe OpenCV [12] elaborat de Radkowski și Wassmann, descris în [9]. Spre deosebire de prototipul din lucrare toate interacțiunile sunt realizate prin mâini, fără ajutorul marcajelor, gesturile recunoscute fiind: pumn, mână strânsă, mână deschisă, degetul arătător, glisarea mâinii. Metaforele de interacțiune descrise sunt: selecție obiect, manipulare obiect (scalare, rotație, translație), asamblare obiect. Trasarea scenei precum și asamblarea obiectelor virtuale se face prin intermediul bibliotecii OpenSceneGraph [13] ce integrează motorul de trasare și un motor fizic, prin care se face alipirea părților obiectelor. Deși cerințele hardware sunt mai ridicate față de prototipul propus ansamblul pune la dispoziție o interacțiune mai intuitivă și familiară utilizatorului.

Prototipurile și sistemul de interacțiune sumarizate până acum, reprezintă o cărămidă pentru interfețelor utilizator tangibile augmentate (*Tangible Augmented Reality*) - TAR. Ele introduc noi principii de proiectare a interfețelor precum: 1. folosirea elementelor fizice pentru a manipula conținutul virtual; 2. interacțiunea să fie continuă în timp și spațiu; 3. suport pentru gesturi folosind mai multe puncte de contact pe suprafață (*multiple touch gestures*); 4. suport pentru potrivirea constrângerilor fizice ale obiectelor, la cerințele sarcinii; 5. colaborare între mai mulți participanți.

Un exemplu de interfață ce folosește această metaforă este în sistemul VOMAR elaborat în [4]. Aici se folosește o padelă pentru a selecta și mișca mobilă virtuală într-un cadru real. Mișcări ale padelei sunt asigurate unor comenzi de tip gesturi, precum înclinarea padelei pentru a plasa un obiect în scenă, sau lovirea obiectului pentru a-l șterge.

DESCRIEREA PROTOTIPULUI

Prototipul este alcătuit din trei componente software: ARToolkit, prin intermediul căruia determinăm locația marcajului în scenă; componenta de captare, care determină începutul și sfârșitul căii (*stroke*) și componenta de recunoaștere, ce face potrivirea traseului înregistrat de cea din urmă componentă cu un set de șabloane pre-stocate.

ARToolkit

El reprezintă o bibliotecă software scrisă în C care folosește tehnici de procesare a imaginilor și viziune pe calculator (*computer vision*) pentru a calcula poziția și orientarea camerei relativ față de un marcaj cu o bordură

pătrată. În suprafața delimitată de această bordură se regăsesc diferite forme ce reprezintă trăsăturile marcajului. Într-o primă instanță, se caută și se regăsesc aceste șabloane prin binarizare și detecție colțuri. Apoi se face o normalizare pentru a găsi corespondențe între trăsăturile din spațiul imagine captat și cel obiect (marcajul din șablon). Pe baza acestor corespondențe între pozițiile colțurilor dintre cele două spații se generează o homografie (*homography*) între cadre succesive. Din aceasta se extrage matricea de transformare care este aplicată obiectelor virtuale. Pentru detalii legate de algoritmul de calculul al acestei matrice se poate consulta lucrarea [2]. O vedere de ansamblu asupra procesului folosit de ARToolkit este ilustrat în Figura 1.

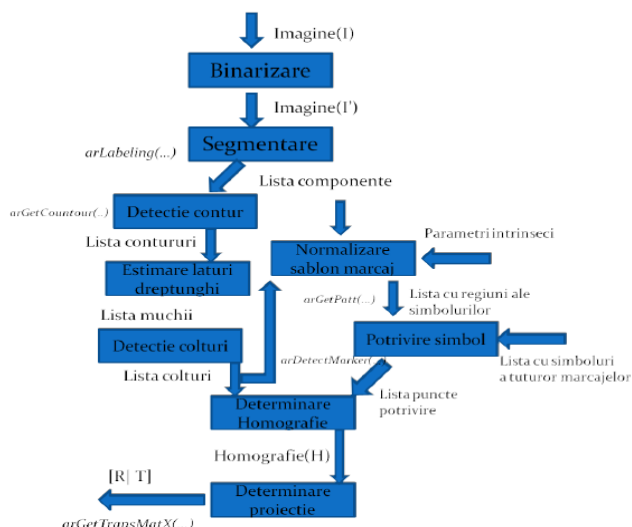


Figura 1. ARToolkit: vedere ansamblu

Motivul alegerii acestei biblioteci în favoarea altora, a fost construcția sa modulară precum și ușurința integrării acesteia pe verticală. Această decompoziție a permis delimitarea funcțiilor necesare unei aplicații AR și studiul în detaliu doar a elementelor relevante scopului lucrării. Suportul său pentru captarea video este expus printr-un API intuitiv și ușor de folosit fiind o interfațare utilă peste sistemul de operare. Componenta inclusă de trasare grafică (*rendering*) pune la dispoziție funcții uzuale care elimină apelurile uzuale, generice, OpenGL (*boilerplate code*). Un aspect negativ este acela că modulul nu a fost actualizat să facă uz de ultimele specificații OpenGL disponibile în plăcile video. Independența față de sistemul de fișiere este un atu care îl face popular nu numai pentru aplicații desktop sau mobile ci și celor destinate în aer liber prin definiții de funcții pentru inițializarea și folosirea modulelor montate pe cap (*Head Mounted Display*)-HMD.

Un alt motiv pentru care s-a optat pentru această bibliotecă a fost viteza ridicată de procesare ce poate susține rate de cadre interactive împreună cu acuratețea determinării poziției marcajului. Deși modelul de marcaje folosit prin structura sa permite estimarea tuturor celor șase grade de libertate (*6-Degree Of Freedom*) -6-DOF, faptul că parametrii extrinseci sunt raportați la un model al camerei deformat, pot duce la erori care se propagă în cazul unei

calibrări necorespunzătoare sau a unui model de calibrare insuficient de bun. Precizia bibliotecii a fost studiată de către Piekarski și Malbezin în [7]. Conform lor, o distanță de 1.5 m și un marker de 2 cm produc erori acceptabile (± 18 mm) pentru asemenea aplicații.

Componenta de captare

Pentru o interacțiune cât mai cufundată în realitate, este necesară delimitarea căii ce va fi recunoscută, de alte interferențe precum: eșuarea pe termen scurt a urmării, ocluziuni sau rotații bruște ce nu pot fi captate. Componenta a fost implementată printr-un automat de stări finite ilustrat în Figura 2.

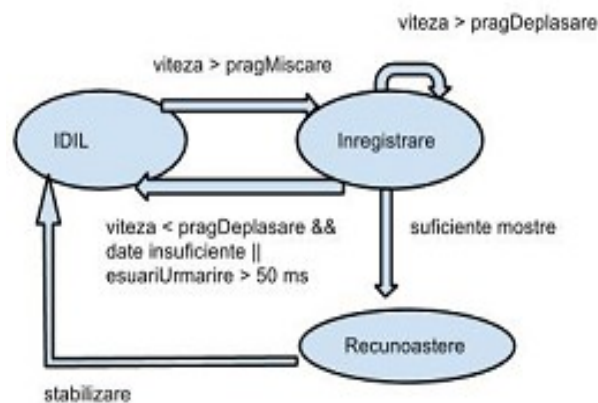


Figura 2. Automatul de stări finite

Se menține o viteză instantanee calculată din pozițiile marcajului de-a lungul unor seturi de cadre dintr-o fereastră de timp de 200 ms. Rata de cadre este estimată pe toată durata rulării aplicației prin intermediul apelurilor unor ceasuri puse la dispoziție de ARToolkit. Trecerea din idil în recunoaștere se face când avem deplasări bruște detectabile. Dacă nu putem detecta aceste deplasări (pentru acel cadru pozițiile marcajului sunt invalide) de-a lungul mai multor cadre vom tranzita în idil. Iar dacă detectăm o încetinire a deplasării și nu am acumulat suficiente mostre vom tranzita din nou în idil. Pe durata recunoașterii vom stoca pozițiile marcajului, iar când avem destule mostre, echivalentul a 1-2 secunde, trimitem rezultatul spre recunoaștere. După o perioadă de 2 secunde sau la detecția opririi mișcării, caracterizată printr-o viteză instantanee scăzută, vom tranzita în starea idil.

Componenta de recunoaștere

Algoritmul folosit este cel descris de Wilson et al. în [10] intitulat *one dollar*. Acesta face potrivirea formelor geometrice 2D definite ca o mulțime ordonată de puncte. O formă candidat la intrare este comparată pe rând cu toate formele de referință care trebuie recunoscute. Acestea din urmă trebuie să se regăsească în memorie, fie prin antrenare fie prin încărcarea dintr-un fișier. În urma fiecărei comparații dintre un candidat și o referință va rezulta un scor de similitudine. Scorul este obținut pe baza distanței euclidiene. Algoritmul este alcătuit din patru pași: 1. Distribuie punctele astfel încât mostrele și referințele să fie de lungime egală. Două forme datorită naturii de generare nu pot avea același număr de puncte. Pentru a compensa acest lucru, când se calculează

distanța, ele trebuie aduse la un număr comun de puncte. Pentru fiecare formă se calculează lungimea ei și se împarte la numărul comun de puncte stabilit. Folosind incrementul rezultat, se parcurge forma, iar în locurile unde distanța dintre puncte consecutive depășește acest increment, noi puncte sunt inserate prin interpolare; 2. Calculează unghiul (unghiul indicator) dintre centrul figurii și primul punct al ei și efectuează o rotație în sensul acelor de ceasornic cu acel unghi. Măsura reprezintă o euristică care aproximează rotația optimă pentru care putem avea o potrivire. Aceasta compensează posibilele rotații ale formei candidat; 3. Se scalează și translatează punctele cu referire la lungimea diagonalei unui pătrat comun pentru toate formele. Operația este necesară pentru ca schimbările în distanțe dintre puncte să fie dependente doar de rotațiile ce vor fi făcute în etapa 4; 4. Găsirea unghiului pentru care distanța dintre mostră și referință este minimă. Aceasta corespunde găsirii acelei rotații a formei de referință pentru a se potrivi cât mai bine mostrei. Pornind de la unghiul indicator se face o căutare într-un interval vecin cu un anumit pas de iterație. Detalii în plus despre funcționare pot fi găsite în [10].

EXPERIMENTE

În scenariile implementate s-a urmărit evaluarea prototipului din punct de vedere: al ratei de eroare în detecția formelor (E_p) pre-stocate și respingerea (E_r) a celor necunoscute; al calității interacțiunii (CI).

$$E_p = \frac{\text{Nr. gesturi fals response}}{\text{Nr. total}}, E_r = \frac{\text{Nr. gesturi fals acceptate}}{\text{Nr. total}}$$

Corectitudinea unei forme este dată de un evaluator uman care apreciază vizual rezultatul căii utilizatorului. S-a îmbunătățit scorul dat de componenta de recunoaștere prin următoarea măsură de încredere:

$$\frac{1}{S_1} (S_1 - \sum_{i=2}^k S_i)$$

unde S_1 este scorul cel mai bun (mare) iar restul sunt următoarele $k-1$ scoruri cele mai bune. O valoare mică, indică posibilitatea confundării unui candidat cu mai multe referințe. De obicei acest comportament este aplicabil candidaților ce nu fac parte din mulțimea referințelor. Calitatea interacțiunii reprezintă diferența dintre 1 și raportul dintre numărul de gesturi nerecunoscute când utilizatorul intenționa trasarea lor și erau corect generate, și numărul total de gesturi executate. Gesturile pre-înregistrate sunt date în Figura 3:



Figura 3. Gesturi pre-înregistrate

Fiecare utilizator a avut o perioadă de antrenare în care aplicația i-a înregistrat aceste gesturi. Scopul a fost familiarizarea cu aplicația și adaptarea gestului la specificul utilizatorului. Scenariul de test constă într-un meniu în care există opțiuni corespunzătoare gesturilor din figură, prin care parcurgem o colecție de animații scurte cu posibilitate de oprire, repornire, trecere la următorul, la precedent, ieșire, resetare. Comenzile trebuiau repetate într-o anumită ordine, fiecare de un anumit număr de ori, timp de 5 minute. Prototipul a fost conceput ca o aplicație desktop pentru o dezvoltare mai rapidă. Din cauza modelului de calibrare supus erorii precum și a calității slabe a camerei integrate, nu s-au putut obține rate interactive folosind marcaje de latură 2 cm precum era indicat de Piekarski și Malbezin [7] ci doar folosind marcaje de 4 cm. Distanța față de cameră era de 1-1.5 m pentru a avea grad de libertate mai ridicat și erori mici în estimarea pozițiilor. Pentru algoritmul *one dollar* s-a ales eșantionarea pozițiilor astfel încât să avem șabloane de 128 de puncte. Intervalul de căutare a unghiului de rotație optim din pasul 4 este de +/- 45° având ca mijloc unghiul indicator, iar pasul de iterație este de 2°, cum au fost folosiți în lucrarea originală. Un candidat este respins dacă gradul de încredere este mai mic de 65% conform măsurii de încredere definite.

REZULTATE ȘI CONCLUZII

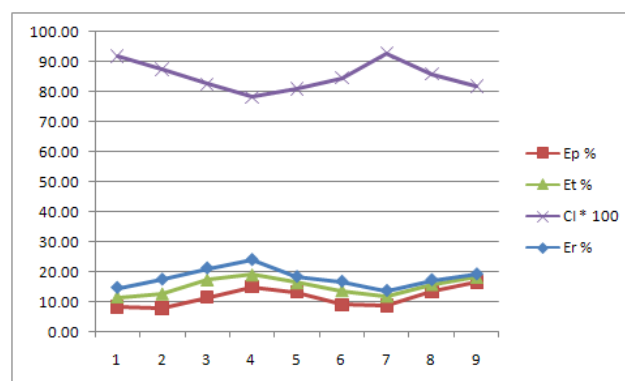


Figura 4. Evoluția erorilor.

Figura 4 rezumă performanțele prototipului propus în contextul de testare amintit. E_r semnifică eroarea totală a prototipului sumând ponderat cele două erori. Ca observații empirice, s-a constatat recunoașterea cu succes a gesturilor de glisare spre dreapta și stânga, în ciuda similitudinii acestora dacă ne gândim că algoritmul este invariant la rotații. Motivul vine din faptul că utilizatorii și-au înregistrat aceste gesturi, iar în consecință ele nu erau line ci erau perturbate de tremurat și eșec în capturare continuă. Aceste mici diferențe au reieșit în calculul distanțelor. S-a încercat folosirea gesturilor din Figura 3 neadaptate la utilizatori și s-a constatat creșterea erorii de recunoaștere. Pe durata testării s-a constatat că multe erori la respingere erau date de dificultatea reproducerii unor forme precum “]” dată fiind logica rigidă de capturare și eșecurile ocazionale de estimare a pozițiilor marcajelor. Acest fapt indică insuficiența stărilor modelate pentru a capta toate stările intermediare și ascunse ce pot apărea la

urmărire. De asemenea explică și rezultatele CI. Acest fapt arată că există în cadrul acestui prototip o curbă de învățare pe care utilizatorii trebuie să o parcurgă pentru a deveni eficienți în interacționarea cu el. Deși rezultatele sunt diferite pentru fiecare utilizator se poate identifica un interval în care variază. O direcție de cercetare este folosirea unui ghidaj (virtual) sau suport care să ușureze definirea gesturilor.

În lucrare s-a prezentat un prototip ce realizează interacțiunea cu utilizatorul într-un mediu AR. Prototipul este capabil să distingă gesturi simple folosite în interfețele dispozitivelor mobile actuale. Limitările prezente vin din partea automatului de stare care nu poate capta toate cazurile ce apar într-un astfel de mediu, precum și a modelului de mișcare folosit care se orientează după maxime și minime locale de-a lungul unui interval scurt de timp. O direcție de studiu este folosirea fluxului optic pe regiunea pătratică delimitată de marker pentru a combate aceste efecte.

MULȚUMIRI

Autorul mulțumește domnului profesor dr. ing. Dorian Gorgan de la Universitatea Tehnică Cluj-Napoca, Facultatea de Automatică și Calculatoare pentru suportul și îndrumarea acordată în realizarea acestei lucrări.

REFERINȚE

1. Azuma, R., Baillot, Y: Recent Advances in Augmented Reality, IEEE Computer Graphics and Applications, 2001
2. Daniel, F., Davis, S.: Model Based Object Pose in 25 Lines of Code, International Journal of Computer Vision, 1995
3. Dobler, D., Haller, M.: ASR – Augmented Sound Reality, ACM SIGGRAPH, 2002
4. Kato, H., Billinghurst, M.: Virtual object manipulation on a table-top AR environment,

Proceedings of the International Symposium on Augmented Reality, 2000

5. Mason, P., Billinghurst, M.: MagicMouse: an Inexpensive 6-Degree-of-Freedom Mouse, Proceedings of Graphite, 2003
6. Piekarski, W.: Using ARToolkit for 3D hand position tracking in mobile outdoor environments, Augmented Reality Toolkit, The First IEEE International Workshop, 2002
7. Piekarski, W., Malbezin, P.: Measuring ARToolkit Accuracy in Long Distance Tracking Experiments, Augmented Reality Toolkit, The First IEEE International Workshop, 2002
8. Radkowski, R. Stritzke, C.: Interactive Hand Gesture-based Assembly for Augmented Reality Applications, Advances in Computer-Human Interactions 2012.
9. Radkowski, R. Wassmann, H.: Using Computer Vision for Utilizing the Human Hand in an Augmented Reality Application, Proceedings of IADIS Computer Graphics and Visualization, 2007
10. Wilson, A. D., Li, Y.: Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes, Proceedings of the ACM Symposium on User Interface Software and Technology, 2007
11. ***, Situl bibliotecii ARToolkit, <http://www.hitl.washington.edu/artoolkit/>
12. ***, Situl bibliotecii de recunoastere de imagini si obiecte folosind calculatorul OpenCV, <http://opencv.willowgarage.com/wiki/>
13. ***, Situl bibliotecii de trasare grafurilor de scenă OpenSceneGraph, <http://www.openscenegraph.org/>