

# Robust Real-Time Visual Odometry Estimation for 3D Scene Reconstruction

Joo-Hee Kim<sup>†</sup> · In-Cheol Kim<sup>\*\*</sup>

## ABSTRACT

In this paper, we present an effective visual odometry estimation system to track the real-time pose of a camera moving in 3D space. In order to meet the real-time requirement as well as to make full use of rich information from color and depth images, our system adopts a feature-based sparse odometry estimation method. After matching features extracted from across image frames, it repeats both the additional inlier set refinement and the motion refinement to get more accurate estimate of camera odometry. Moreover, even when the remaining inlier set is not sufficient, our system computes the final odometry estimate in proportion to the size of the inlier set, which improves the tracking success rate greatly. Through experiments with TUM benchmark datasets and implementation of the 3D scene reconstruction application, we confirmed the high performance of the proposed visual odometry estimation method.

**Keywords :** RGB-D Images, Visual Odometry, 3D Scene Reconstruction, Feature-Based Sparse Method

## 3차원 장면 복원을 위한 강건한 실시간 시각 주행 거리 측정

김 주 희<sup>†</sup> · 김 인 철<sup>\*\*</sup>

### 요 약

본 논문에서는 RGB-D 입력 영상들로부터 3차원 공간을 움직이는 카메라의 실시간 포즈를 효과적으로 추적할 수 있는 시각 주행 거리 측정기를 제안한다. 본 논문에서 제안하는 시각 주행 거리 측정기에서는 컬러 영상과 깊이 영상의 풍부한 정보를 충분히 활용하면서도 실시간 계산량을 줄이기 위해, 특징 기반의 저밀도 주행 거리 계산 방법을 사용한다. 본 시스템에서는 보다 정확한 주행 거리 추정치를 얻기 위해, 카메라 이동 이전과 이동 이후의 영상에서 추출한 특징들을 정합한 뒤, 정합된 특징들에 대한 추가적인 정상 집합 정제 과정과 주행 거리 정제 작업을 반복한다. 또한, 정제 후 잔여 정상 집합의 크기가 충분히 않은 경우에도 잔여 정상 집합의 크기에 비해 최종 주행 거리를 결정함으로써, 추적 성공률을 크게 향상시켰다. TUM 대학의 벤치마크 데이터 집합을 이용한 실험과 3차원 장면 복원 응용 시스템의 구현을 통해, 본 논문에서 제안하는 시각 주행 거리 측정 방법의 높은 성능을 확인할 수 있었다.

**키워드 :** RGB-D 영상, 시각 주행 거리 측정, 3차원 장면 복원, 특징-기반 저밀도 방법

### 1. 서 론

최근 들어 컴퓨터 비전, 지능 로봇, 모바일 컴퓨팅 분야를 중심으로, 카메라의 실시간 입력 영상으로부터 3차원 공간 모델을 자동으로 생성하기 위한 3차원 장면 복원(3D scene reconstruction) 기술에 관한 연구가 활발하다. 효과적인 3차

원 장면 복원을 위해서는 다양한 요소 기술들이 필요하지만, 그중에서도 입력 영상들을 분석하여 3차원 공간에서 카메라의 실시간 이동 거리와 위치를 추정하는 시각 주행 거리 측정(visual odometry estimation) 기술이 매우 중요하다. 기존 연구들에서 소개된 시각 주행 거리 측정 방법들은 어떤 입력 영상을 이용하느냐에 따라 크게 Fig. 1의 (a)와 같은 깊이 영상(depth image)을 이용하는 방법과 Fig. 1의 (b)와 같은 컬러 영상(color image)을 이용하는 방법, 그리고 깊이 영상과 컬러 영상을 모두 이용하는 방법, 즉 RGB-D 영상을 이용하는 방법으로 나누어볼 수 있다. 또한, 기존의 방법들은 입력 영상의 모든 픽셀들을 거리 측정에 이용하느냐, 그중 일부분만을 이용하느냐에 따라 고밀도 방법(dense method)과 저밀도 방법(sparse method)로도 나누어볼 수 있다.

※ 본 연구는 경기도의 경기도지역협력연구센터사업의 일환으로 수행하였음.  
※ 이 논문은 2014년도 한국정보처리학회 추계학술발표대회에서 'RGB-D 영상을 이용한 강건한 실시간 시각 주행 거리 측정'의 제목으로 발표된 논문을 확장한 것임.

† 준 회원: 경기대학교 컴퓨터과학과 석사과정

\*\* 종신회원: 경기대학교 컴퓨터과학과 교수

Manuscript Received: January 27, 2015

Accepted: February 24, 2015

\* Corresponding Author: In-Cheol Kim(kic@kyonggi.ac.kr)

깊이 영상의 경우, 상대적으로 성능이 낮은 저밀도 방법 대신 ICP(Iterative Closest Point) 기반의 고밀도 방법이 주로 사용되고 있다. 이 방법은 카메라 이동 이전과 이동 이후의 두 깊이 영상의 모든 픽셀을 3차원 공간에 역투영(back projection)하여 포인터 클라우드(point cloud)를 생성하고, 이렇게 생성된 두 개의 포인터 클라우드를 서로 정합해봄으로써 카메라의 모션, 즉 주행 거리를 알아내는 방법이다[1]. 이와 같이 깊이 영상만을 이용하는 주행 거리 측정 방법은 깊이 변화가 별로 없는 벽면이나 복도와 같은 공간들에서는 주행 거리 측정이 어렵다는 문제점을 가지고 있다. 반면에, 깊이 영상에 비해 정보가 풍부한 컬러 영상을 이용하는 경우에는, 저밀도 방법만으로도 높은 측정 성능을 얻을 수 있어, 계산 요구량이 많은 고밀도 방법보다 저밀도 방법이 더 폭넓게 사용되고 있다. 컬러 영상을 저밀도로 이용하는 대표적인 방법은 두 컬러 영상들로부터 각각 독립적으로 SIFT, SURF, FAST 등과 같은 특징들(features)을 추출해낸 뒤, 두 영상의 특징들만을 서로 정합해봄으로써 두 영상 간의 카메라 주행 거리를 계산해낸다[2].

최근에는 깊이 영상과 컬러 영상을 주행 거리 계산에 모두 이용하려는 소위 RGB-D 영상 기반의 주행 거리 측정 방법들이 활발히 연구되고 있다. RGB-D 영상을 이용하는 고밀도 방법에서는 카메라 이동 이전 영상의 모든 픽셀들을 이동 이후의 영상 위에 재투영(reprojection)시킨 뒤, 각 픽셀의 광도 측정 에러(photometric error)를 최소화할 수 있는 카메라 주행 거리를 계산해낸다[3]. 하지만 이와 같은 고밀도 방법은 계산 요구량이 많아 GPU와 같은 특수한 하드웨어의 도움 없이는 실시간 카메라 주행 거리 측정기로 이용하기 어렵다는 문제점이 있다.



(a) Depth Image (b) Color Image

Fig. 1. RGB-D Input Images from Camera

본 논문에서는 3차원 공간에서 6자유도(DOF, Degree of Freedom)로 움직이는 카메라의 실시간 포즈를 추적하기 위해, RGB-D 영상 기반의 저밀도 시각 주행 거리 측정 방법을 제안한다. 이 방법은 높은 주행 거리 측정 성능과 실시간성을 확보하기 위해, 깊이 영상과 컬러 영상을 모두 사용하지만 각 영상에서 추출된 특징들 위주로 비교 연산을 제한하는 저밀도 방법을 사용한다. 또한, 본 시스템에서는 정확도 향상을 위해, 정합된 특징점들에 대한 추가적인 정상 집합 정제(inlier set refinement) 과정과 이들을 이용한 주행 거리 정제 작업을 받

복한다. 본 논문에서는 제안하는 시각 주행 거리 측정기의 성능을 분석하기 위해, TUM (Technische Universität München) 대학에서 개발한 벤치마크(benchmark) 데이터 집합들[4]을 이용한 정량적 실험 결과와 3차원 장면 복원 결과를 소개한다.

## 2. 카메라 모션과 시각 주행 거리 측정

본 논문에서는 3차원 공간에서 6자유도(DOF)로 움직이는 카메라의 실시간 포즈와 주행 거리를 구하고자 한다. 이러한 카메라의 움직임은 3차원 유클리드 공간(Euclidean space) 안의 강체 모션(rigid body motion)으로 볼 수 있으며, 따라서 강체를 구성하는 임의의 두 점  $p$ 와  $q$  사이의 거리(distance)와 회전(orientation)을 유지하는 Equation (1)과 같은 변환 함수(transformation function)  $g$ 로 표현할 수 있다.

$$g: R^3 \rightarrow R^3; x \mapsto g(x) \tag{1}$$

즉,  $\|p - q\| = \|g(p) - g(q)\|$ ,  $g(p) \times g(q) = g(p \times q)$ 를 만족한다.

이러한 3차원 유클리드 공간 안에서의 강체 모션은 다시 회전 운동(rotational motion) 부분과 병진 운동(translational motion) 부분으로 나눌 수 있으며, 회전 운동 부분은 방향(orientation)의 변화를, 병진 운동 부분은 위치(position)의 변화를 각각 나타낸다. 따라서 일반적으로 3차원 유클리드 공간 안에서의 강체 모션은 회전 운동을 위한 3자유도와 병진 운동을 위한 3자유도를 합쳐, 총 6자유도를 갖는다. 회전 운동을 나타내는 다양한 표현법들이 있으나, 가장 보편적인 방법은 3x3 회전 행렬  $R$ (rotation matrix)로 표현하는 것이다. 반면에, 병진 운동은 각각 x축, y축, z축 방향으로의 이동거리를 나타내는 3차원 병진 벡터(translation vector)  $t \in R^3$ 로 표현하는 것이 일반적이다. 따라서 3차원 유클리드 공간 안에서의 강체 모션은 회전 운동과 병진 운동의 결합으로서, Equation (2)와 같은 4x4 행렬  $G$ 로 표현하기도 한다.

$$G = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \tag{2}$$

이동 직전의 카메라 포즈가  $p$ 이고 카메라 모션이 함수  $g$ 로 주어진다면, 이동 직후의 카메라 포즈  $p'$ 는 Equation (3)과 같이 행렬 곱으로 계산할 수 있다.

$$p' = g(p) = g(G, p) = G \cdot p \tag{3}$$

본 논문에서는 이동 직전 포즈  $p$ 와 이동 직후 포즈  $p'$ 에서 각각 수집한 영상  $I_p$ 와  $I_{p'}$ 을 이용하여, 3차원 공간을 이동 중인 카메라의 실시간 모션  $g$ 를 행렬  $G$  형태로 계산해내는 것을 시각 주행 거리 측정(visual odometry estimation)이라고 한다.

### 3. 시각 주행 거리 측정기 설계

본 논문에서는 RGB-D 영상으로부터 카메라의 실시간 주행 거리를 효과적으로 측정할 수 있는 특징점 위주의 저밀도 시각 주행 거리 측정기를 제안한다. 본 시스템의 전체 시스템 구조도는 Fig. 2와 같다. 본 시스템은 크게 특징 추출 단계(feature extraction phase), 특징 정합 단계(feature matching phase), 주행 거리 측정 단계(odometry estimation phase)로 이루어진다. 특징 추출 단계에서는 RGB-D 영상을 입력받아 전처리를 수행한 후, FAST-9 코너 검출기와 서술자 계산을 통해 특징을 추출한다. 특징 정합 단계에서는 이전 프레임(previous frame)과 현재 프레임(current frame) 간의 특징들을 비교하여 정합한다. 주행 거리 측정 단계에서는 정상적으로 정합된 특징 집합을 가려내고, 이들로부터 주행 거리를 계산해낸다.

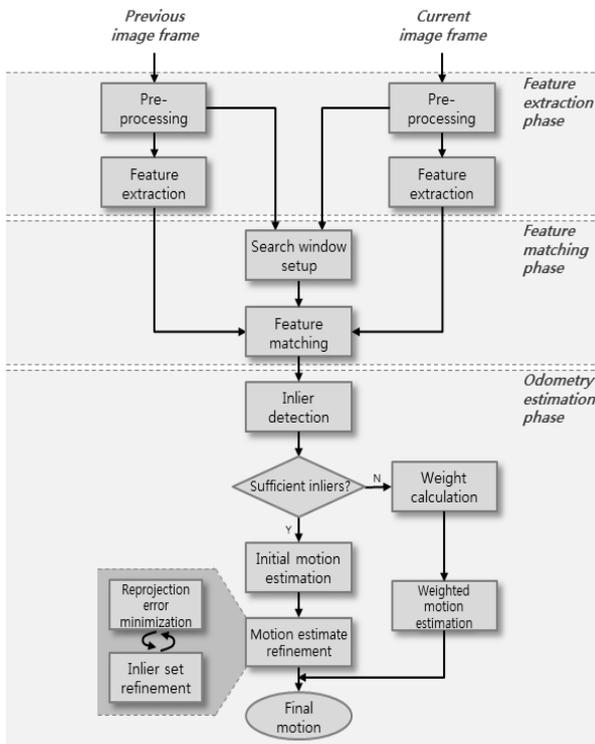


Fig. 2. Overall System Structure

#### 3.1 특징 추출

본 시스템은 먼저 RGB-D 카메라로부터 RGB-D 영상을 입력받아 전처리(preprocessing) 과정을 거친다. 전처리 과정에서는 컬러 영상을 Fig. 3의 (a)와 같은 회색 음영 영상(gray scale image)으로 변환하고, 노이즈의 영향을 줄이기 위해 가우시안 커널로 스무딩(smoothing) 한다. 또한, 크기(scale) 변화에 강인한 특징 추출을 위해, 다운 샘플링(down sampling)을 이용하여 영상을 1/2씩 축소시켜가면서 다양한 크기의 가우시안 피라미드(Gaussian pyramid)를 생성한다.

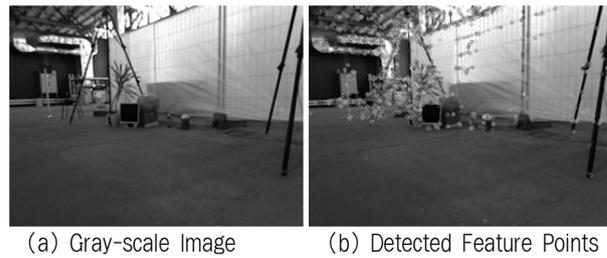


Fig. 3. Image Conversion and Feature Points Detection

전처리 과정이 끝나면 FAST-9 코너 검출기를 사용하여, 가우시안 피라미드의 각 레벨마다 특징점을 검출한다. FAST-9 코너 검출기는 영상의 한 픽셀에 대한 특징점 여부를 판단한다. 영상의 특정 픽셀로부터 거리가 3인 주변의 16개 픽셀을 검사하여, 16개 중 보다 어둡거나 밝은 픽셀이 9개 이상일 경우, Fig. 3의 (b)와 같이 특징점으로 검출한다. 특징점을 검출한 후에는 해당 특징점 위치의 깊이값을 깊이 영상으로부터 추출하고, 대응되는 깊이값이 없는 특징점은 삭제한다. 또한, 특징점의 분포를 균일하게 유지하기 위해 각 피라미드 레벨은 80x80 픽셀의 버킷(bucket)으로 이산화시키고, 각 버킷에는 25개의 특징점들을 저장하도록 한다. 각 특징점에는 특징점 주변의 9x9 픽셀의 광도를 나타내는 80바이트의 서술자가 할당된다. 서술자의 광도는 0을 평균(zero mean)으로 하도록 정규화하고, 가장 아래의 오른쪽 픽셀은 생략한다.

#### 3.2 특징 정합

본 시스템에서는 이전 프레임과 현재 프레임 간의 특징 정합 작업에 앞서 정합 대상의 탐색 범위를 제한할 목적으로, 카메라의 회전 운동량을 미리 추정해본다. 이를 위해 본 시스템에서는 다운 샘플링(down-sampling) 된 두 영상 프레임들을 이용하여 회전 후 픽셀들의 위치 오차를 최소화할 수 있는 회전 운동량을 추정해낸다.



Fig. 4. Feature Matching Between Two Image Frames

두 프레임 간의 특징 정합에는 각 특징별 서술자의 유사도를 비교하는 방법을 사용하여 Fig. 4와 같이 정합한다. 두 프레임으로부터 각각 추출된 두 서술자  $D_a$ 와  $D_b$  사이의 유사도는 Equation (4)와 같이 서술자들의 절대 차의 합(sum-of-absolute differences)을 계산하여 점수화할 수 있다. Equation (4)의  $D$ 는 서술자를 나타내고,  $l$ 은 서술자의 길이를 나타낸다.

$$score(D_a, D_b) = \sum_{i \in I} |D_a(i) - D_b(i)| \quad (4)$$

이와 같이 방식으로 두 프레임에서 추출한 모든 특징 쌍들에 대해 서술자 유사도를 계산하여 그 점수가 가장 낮은 특징들을 서로 정합시킨다. 하지만 만약 어떤 특징 쌍이 미리 추정해놓은 회전 운동량의 범위를 크게 벗어난 경우에는 이상치로 간주하여 정합 대상에서 제외한다.

### 3.3 주행 거리 측정

본 시각 주행 거리 측정 시스템에서는 정상적으로 정합된 특징들을 이용해 주행 거리의 초기 추정치(initial estimate)를 계산한 다음, 정합된 특징점들에 대한 추가적인 정상 집합 정제(inlier set refinement) 과정과 이들을 이용한 주행 거리 정제 작업을 반복함으로써 주행 거리 추정치의 정확도를 향상시킨다. 주행 거리의 초기 추정치는 정합된 특징들 사이의 유클리드 거리(Euclidean distance)의 합을 최소화하는 Horn의 방법을 이용해 계산한다. 정합된 특징점들에 관한 추가적인 정상 집합 정제 작업은 두 특징점의 전역 좌표  $x_{wc}$ 와  $x'_{wc}$  사이의 차를 계산하는 Equation (5)와 같은 판별식을 이용한다.

$$\|x_{wc} - x'_{wc}\| < \delta \quad (5)$$

Equation (5)에서  $x_{wc}$ 와  $x'_{wc}$ 는 각각 이미지 프레임  $a$ 와  $b$ 에서 추출된 특징점의 전역 좌표(world coordinates)를 나타내며, 전역 좌표는 특징점 검출과 함께 깊이 영상으로부터 추출한 깊이값을 이용하여 구할 수 있다. 또한, 두 특징점은 앞서 설명한 특징 정합 과정을 통해 이미 서로 정합된 상태임을 가정한다. Equation (5)에 따라 전역 좌표의 차가 임계값  $\delta$  미만인 특징 쌍들은 정상 집합(inlier set)에 포함되지만, 그렇지 못한 특징 쌍들은 정상 집합에서 제외된다.

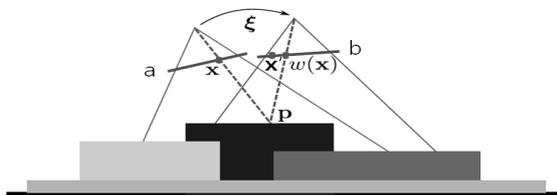


Fig. 5. Reproduction Error Calculation of Feature Points

정상 집합 정제 작업이 완료되면, 정상 집합에 포함된 특징들에 대한 재투영 오차(reproduction error)를 최소화할 수 있도록 주행 거리를 정제하는 작업이 진행된다. Fig. 5에는 정합된 두 특징점  $x$ 와  $x'$ , 그리고  $x$ 를 재투영한 점  $w(x)$  사이의 관계를 나타내고 있다. Fig. 5에서 3차원 공간 표면(surface) 위의 한 점  $p$ 는 깊이 정보를 이용해 2차원 영상 위의 한 특징점  $x$ 를 3차원 공간 위에 역투영(back projection)한  $x$ 의 대응점이다. 이미지 프레임  $a$ 와  $b$  사이에 발생한 카메라 모션이

$\xi$ 이고 이 모션에 따라 점  $p$ 를 이미지 프레임  $b$  위에 재투영한 대응점이  $w(x)$ 라고 하면, 두 특징점  $x$ 와  $x'$  사이의 재투영 오차는  $e(x, x') = \|x' - w(x)\|$ 로 계산할 수 있다. 따라서 두 이미지 프레임  $a$ 와  $b$  사이의 재투영 오차는 Equation (6)과 같이 정상 집합  $S$  내 모든 특징 쌍들에 대한 재투영 오차의 제곱 합(sum of squares)으로 계산된다.

$$E_\xi(a, b) = \sum_{(x, x') \in S} e(x, x')^2 = \sum_{(x, x') \in S} \|x' - w(x)\|^2 \quad (6)$$

두 이미지 프레임 사이의 재투영 오차식  $E_\xi(a, b)$ 을 최소화할 수 있는 최적의 카메라 모션, 즉 주행 거리  $\xi$ 를 찾기 위해, 본 연구에서는 비선형 최소 자승 문제 해결 기법(nonlinear least squares solver)을 이용한다. 그리고 이렇게 구한 주행 거리 추정치  $\xi$ 의 정확도를 더욱 향상시키기 위해서, 재투영 오차가 충분히 작아질 때까지 다시 정상 집합 정제 작업과 이들을 이용한 주행 거리 재계산을 반복한다. 만약 정상 집합의 개수가 정해진 기준치 이하로 감소하여 정확한 주행 거리를 얻기 어려운 경우에는, 정상 집합의 개수에 비례하는 낮은 가중치(weight)를 주행 거리 추정치에 곱한 뒤, 그 결과만큼 최종 주행 거리로 결정한다.

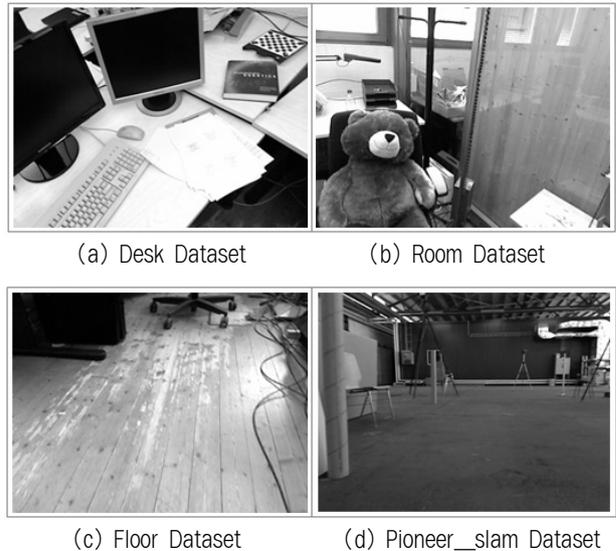


Fig. 6. Datasets for Experiment

## 4. 실험 및 평가

### 4.1 실험

본 논문에서 제안한 시각 주행 거리 측정기(wFVO)의 성능 실험을 위해, Windows 7 64bit, Intel Core @ 3.40GHz 플랫폼에서 visual studio 2010과 C++언어를 이용하여 시각 주행 거리 측정기를 구현하였다. 성능 실험을 위해서 카메라의 실제 주행 거리와 포즈 데이터를 포함하고 있는 TUM 대학의 벤치마크 공개 데이터 집합[7]을 사용하였다. 본 실험에서는 TUM 벤치마크 데이터 집합 중 Desk, Room,

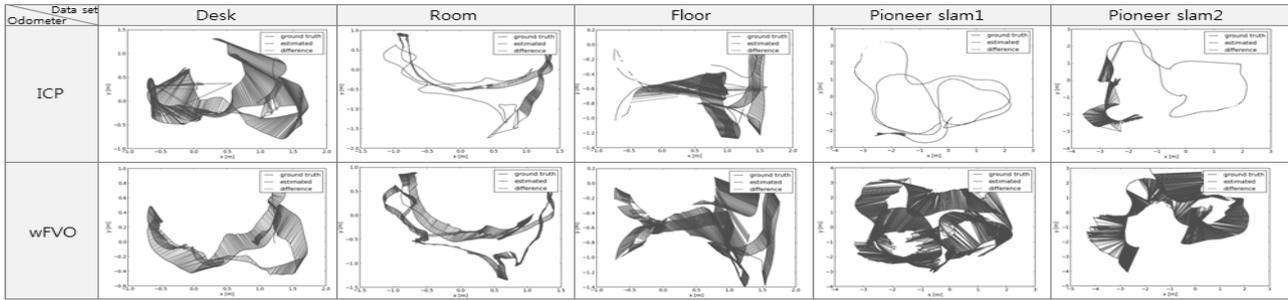


Fig. 7. Estimated Trajectories from Two Different Visual Odometers

Floor, Pioneer\_slam1, Pioneer\_slam2 등 총 5가지의 다양한 특징을 갖는 데이터 집합들을 이용하였다. Desk 데이터 집합과 Room 데이터 집합은 Fig. 6의 (a)와 (b)에서 볼 수 있듯이 각각 여러 물건이 올려져 있는 책상과 여러 가구가 배치되어있는 방 내부를 촬영한 영상으로, 깊이와 컬러 정보가 풍부하다. Floor 데이터 집합은 Room 데이터 집합처럼 방 내부를 촬영하였지만, Fig. 6의 (c)에서 볼 수 있듯이 바닥 위주로 촬영하였기 때문에 다른 영상들에 비해 깊이 변화가 적다. Pioneer\_slam1 데이터 집합과 Pioneer\_slam2 데이터 집합은 Fig. 6의 (d)와 같이 장애물이 적고 넓은 창고 내부를 Pioneer 로봇을 이용하여 촬영한 영상으로, 깊이 정보가 없는 부분도 부분적으로 포함되어있다.

첫 번째 실험에서는 본 논문에서 제안한 RGB-D 영상을 이용하는 저밀도 주행 거리 계산 방법(wFVO)과 깊이 영상만을 이용하는 고밀도 계산 방법(ICP)의 성능을 카메라 포즈 면에서 정성적으로 비교해보았다. 이를 위해 각 방법으로 계산한 주행 거리를 토대로 카메라의 실시간 추정 포즈를 구하고, 이것을 카메라의 실제 포즈(ground truth)들과 비교하였다. Fig. 7은 두 방법으로 구한 카메라의 추정 포즈와 실제 포즈를 비교한 결과를 나타내고 있다. Fig. 7에서 파란색 선은 카메라의 추정 포즈를, 검은색 선은 실제 포즈를 각각 나타내며, 빨간색 선은 두 포즈 사이의 오차 크기를 나타낸다. 본 논문에서 제안한 wFVO의 경우는 대부분의 실험 데이터 집합에서 실제 카메라 포즈와의 오차가 비교적 균일하게 유지될 뿐만 아니라, 전체 이동 경로에 걸쳐 카메라 포즈와 주행 거리의 측정이 끊김 없이 성공적으로 진행되었음을 확인할 수 있다. 반면에, ICP의 경우는 Room, Floor, Pioneer\_slam1, Pioneer\_slam2 등 대부분의 실험 데이터 집합들에서 주행 거리 측정에 실패하여 카메라 포즈를 추정하지 못하였으며, 주행 거리 측정이 가능한 Desk 데이터 집합에서도 wFVO보다 더 큰 오차를 보여주는 등 낮은 성능을 확인할 수 있다.

두 번째 실험에서는 wFVO와 ICP를 주행 거리의 정확도 면에서 정량적으로 비교해보았다. 이를 위해 두 방법으로 계산한 주행 거리의 오차 거리(error distance)를 서로 비교

하였다. Table 1은 각 실험 데이터 집합에 대한 두 주행 거리 측정 방법의 오차 거리를 RMSE(Root Mean Square Error), 평균(Mean), 중간값(Median), 표준편차(Standard Deviation), 최솟값(Min), 최댓값(Max) 등의 다양한 척도로 보여주고 있다. Desk와 Floor 실험 데이터 집합에서는 wFVO의 오차 거리가 더 짧아 ICP에 비해 높은 정확도를 보여주지만, Room이나 Pioneer slam과 같은 데이터 집합에서는 ICP가 wFVO에 비해 더 높은 정확도를 보여준다. 하지만 이것은 ICP가 주행 거리 측정에 성공한 매우 제한적인 경우들에 한해 얻은 결과임을 주목해야 한다.

Table 1. Absolute Trajectory Error(m)

| Data set | Desk      |       | Room      |       | Floor     |       | Pioneer slam1 |       | Pioneer slam2 |       |
|----------|-----------|-------|-----------|-------|-----------|-------|---------------|-------|---------------|-------|
|          | ICP-based | wFVO  | ICP-based | wFVO  | ICP-based | wFVO  | ICP-based     | wFVO  | ICP-based     | wFVO  |
| RMSE     | 0.706     | 0.392 | 0.128     | 0.385 | 0.448     | 0.347 | 0.268         | 1.408 | 0.922         | 1.28  |
| Mean     | 0.657     | 0.37  | 0.098     | 0.337 | 0.339     | 0.289 | 0.241         | 1.271 | 0.833         | 1.225 |
| Median   | 0.708     | 0.407 | 0.078     | 0.269 | 0.232     | 0.199 | 0.218         | 1.248 | 0.866         | 1.271 |
| Std.     | 0.259     | 0.129 | 0.082     | 0.186 | 0.293     | 0.192 | 0.117         | 0.607 | 0.396         | 0.375 |
| Min.     | 0.127     | 0.057 | 0.011     | 0.03  | 0.123     | 0.126 | 0.053         | 0.15  | 0.089         | 0.195 |
| Max.     | 1.228     | 0.608 | 0.849     | 0.812 | 1.975     | 0.835 | 0.514         | 3.245 | 2.507         | 2.002 |

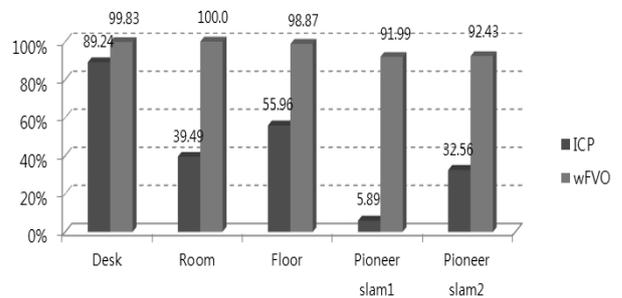


Fig. 8. Success Rate(%)

세 번째 실험에서는 wFVO와 ICP를 주행 거리 측정의 신뢰도 면에서 정량적으로 비교해보았다. 이를 위해 서로 다

른 실험 데이터 집합에 대한 두 방법의 주행 거리 측정 성공률(success rate)을 서로 비교하였다. Fig. 8은 두 방법의 주행 거리 측정 성공률을 비교한 그래프를 나타내고 있다. 백분율로 표현된 주행 거리 측정 성공률을 살펴보면, 모든 실험 데이터 집합들에서 wFVO는 90% 이상의 높은 측정 성공률을 보이는 데 반해, ICP는 Desk 데이터 집합을 제외한 거의 모든 데이터 집합들에서 낮은 성공률을 보여주고 있다. 특히 Pioneer slam1과 같은 경우, ICP는 5.89%의 매우 낮은 성공률을 보였으나, wFVO는 91.99%의 높은 성공률을 보였음을 알 수 있다. 이와 같은 실험 결과를 통해, 본 논문에서 제안한 wFVO의 높은 신뢰도와 안정성을 확인할 수 있었다.

4.2 응용

본 논문에서는 본 연구의 시각 주행 거리 측정기(wFVO)와 pcl 라이브러리(point cloud library)에서 제공하는 Kinect fusion large scale 시스템을 사용하여, 3차원 장면 복원 시스템(3D scene reconstruction system)을 구현하였다.

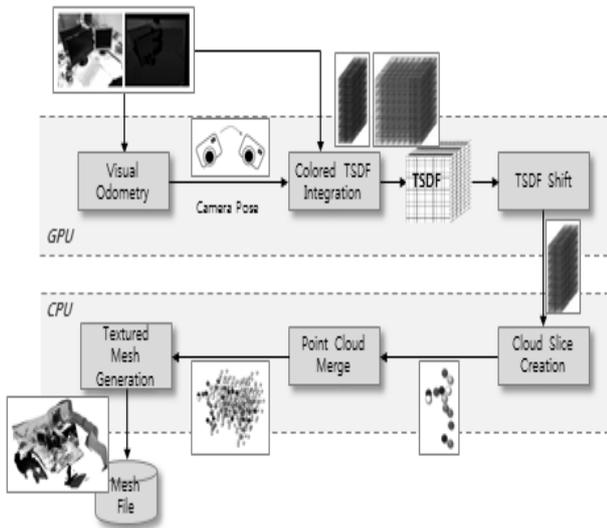


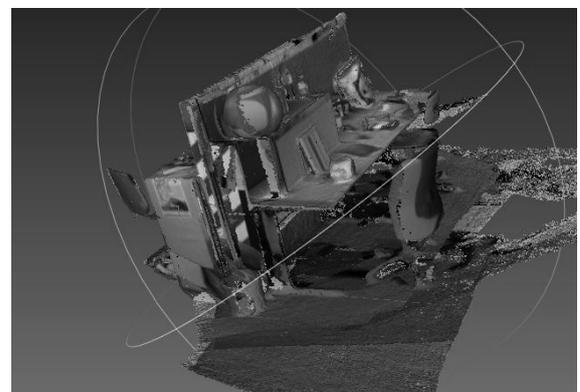
Fig. 9. 3D Scene Reconstruction System

Fig. 9는 시각 주행 거리 측정기를 이용하여 구현한 3차원 장면 복원 시스템의 개략적인 구조도이다. 3차원 장면 복원을 위해서는 먼저 RGB-D 영상으로부터 카메라의 주행 거리와 실시간 포즈(camera pose)를 측정한다(Visua Odometry). 그리고 실시간 카메라 포즈를 이용하여, 새로운 입력 영상으로부터 3차원 공간 모양을 나타내는 TSDF(Truncated Surface Distance Function) 큐브 모델을 계산해낸다. 그리고 이것을 기존의 TSDF 큐브 모델에 결합함으로써 통합된 3차원 공간 모델을 얻어낸다(Colored TSDF Integration). TSDF 큐브 모델이란, 잡음과 불확실성을 내포한 깊이 영상으로부터 3차원 공간 표면을 효과적으로 표현하기 위해, 전

체 공간을 일정한 크기의 정육면체 복셀(voxel)들로 구성된 커다란 하나의 큐브(cube)로 표현하고, 각 복셀에는 물체 표면과의 거리를 나타내는 TSDF값과 그 값의 신뢰도를 나타내는 가중치(weight)를 함께 저장하는 방식을 말한다. 본 시스템에서는 3m\*3m\*3m 크기의 공간을 표현할 수 있는 하나의 TSDF 큐브를 GPU 내에 저장하며, 새로운 입력 영상이 들어올 때마다 이 영상에서 추출한 새로운 공간 정보를 TSDF 큐브로 표현한 다음, 이것을 이전 영상들로부터 구축해놓은 TSDF 큐브에 결합함으로써 TSDF 큐브 모델을 갱신해나간다. 만약 카메라가 3m\*3m\*3m 크기의 기존 공간 제한을 벗어나 새로운 영역으로 이동하게 되면, GPU 메모리에 저장하고 있던 TSDF 큐브 모델 중 일부는 CPU가 관리하는 RAM 기억장소로 이동하여(TSDF Shift), 점군(point cloud) 데이터 형태로 저장된다(Cloud Slice Creation). RAM 기억장소로 옮겨온 점군 슬라이스는 광역 공간을 나타내는 전역 점군(global point cloud) 데이터 집합에 결합된 후(Point Colud Merge), 가시성을 높이기 위해 메쉬 파일(mesh file)로 변환된다(Textured Mesh Generation).



(a) Real scene



(b) Reconstructed scene

Fig. 10. Example of 3D Scene Reconstruction (1) : Desk



(a) Real Scene



(b) Reconstructed Scene

Fig. 11. Example of 3D Scene Reconstruction (2) : Room

Fig. 10과 Fig. 11은 각각 TUM 벤치마크 데이터 집합에 포함된 Desk 장면과 본 대학의 실험실 Room 장면에 대해, 3차원 장면 복원 시스템을 수행한 결과를 나타낸다. 그림에서 좌측 (a)는 실제 장면을 나타내고, 우측 (b)는 본 시스템을 이용하여 복원한 3차원 모델을 나타낸다. 우리는 실제 장면에 매우 유사한 복원 결과를 확인할 수 있다. 이와 같은 높은 품질의 복원 결과를 얻을 수 있는 데에는 본 논문에서 제안한 실시간 시각 주행 거리 측정기 wFVO의 높은 신뢰도와 정확도가 큰 역할을 한 것으로 판단한다.

## 5. 결 론

본 논문에서는 3차원 장면 복원을 위해, RGB-D 입력 영상들로부터 카메라의 실시간 포즈를 효과적으로 추적할 수 있는 시각 주행 거리 측정기를 제안하였다. 본 논문에서 제안한 시각 주행 거리 측정기에서는 컬러 영상과 깊이 영상의 풍부한 정보를 충분히 활용하면서도 실시간 계산량을 줄이기 위해, 특징 기반의 저밀도 주행 거리 계산 방법을 사용한다. 본 시스템에서는 보다 정확한 주행 거리 추정치를 얻기 위해, 카메라 이동 이전과 이동 이후의 영상에서 추출한 특징들을 정합한 뒤, 정합된 특징들에 대한 추가적인 정상 집합 정제 과정과 주행 거리 정제 작업을 반복한다. 또한, 정제 후 잔여 정상 집합의 크기가 충분치 않은 경우에도 잔여 정상 집합의 크기에 비례해 최종 주행 거리를 결정함으로써, 추적 성공률을 크게 향상시켰다. 한편, 대규모 광역 공간에 대한 3차원 장면 복원 시에는 시각 주행 거리 측정기의 작은 오차도 카메라 이동이 계속됨에 따라 큰 오차로 누적되어 나타날 수

있다. 이러한 문제를 극복하기 위해 컴퓨터 비전과 지능 로봇 분야에서는 다양한 센서와 공간 정보를 활용하여 카메라 이동 궤적을 바르게 수정하는 포즈 SLAM 기술에 관한 연구가 활발히 진행되어오고 있다. 향후에는 이러한 포즈 SLAM 기술을 적용하여 본 주행 거리 측정 시스템을 확장하는 연구도 의미 있는 작업이 될 것으로 기대한다.

## References

- [1] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, and P. Kohli, et al., "KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera," Proc. of 24th Annual ACM Symp. on User Interface Software and Technology, pp.559-568, 2011.
- [2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, Vol.29, No 6, pp.1052-1067, 2007.
- [3] C. Kerl, J. Sturm, and D. Cremers, "Robust Odometry Estimation for RGB-D Cameras," Proc. of IEEE Intl. Conf. on Robotics and Automation(ICRA), pp.3748-3754, 2013.
- [4] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera," Proc. of Intl. Symp. on Robotics Research(ISRR), 2011.
- [5] T. Whelan, M. Kaess, M. F. Fallon, H. Johannsson, J. J. Leonard, and J. B. McDonald, "Kintinuous: Spatially Extended KinectFusion," Proc. of RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras, 2012.
- [6] F. Steinbrucker, C. Kerl, and D. Cremers. "Large-Scale Multi-Resolution Surface Reconstruction from RGB-D Sequences," Proc. of IEEE Intl. Conf. on Computer Vision (ICCV), 2013.
- [7] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM System," Proc. of Intl. Conf. on Intelligent Robot System (IROS), 2012.
- [8] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust Real-Time Visual Odometry for Dense RGB-D Mapping," Proc. of IEEE Intl. Conf. on In Robotics and Automation(ICRA), pp.5724-5731, 2013.
- [9] F. Steinbrucker, J. Sturm, and D. Cremers, "Real-Time Visual Odometry from Dense RGB-D Images," Proc. of IEEE Intl. Conf. on Computer Vision Workshops(ICCV Workshops), pp. 719-722, 2011.
- [10] H. Silva, E. Silva, and A. Bernardino, "Combining Sparse and Dense Methods in 6D Visual Odometry," Proc. of IEEE Intl. Conf. on Autonomous Robot Systems(Robotica), pp.1-6, 2013.
- [11] M. Nowicki, P. Skrzypczynski, "Combining Photometric and Depth Data for Lightweight and Robust Visual Odometry," Proc. of IEEE European Conf. on Mobile Robots(ECMR), pp.125-130, 2013.



**김 주 희**

e-mail : jh.k@kyonggi.ac.kr  
2014년 경기대학교 컴퓨터과학과(학사)  
2014년~현재 경기대학교 컴퓨터과학과  
석사과정  
관심분야: 인공지능, 컴퓨터 비전



**김 인 철**

e-mail : kic@kyonggi.ac.kr  
1985년 서울대학교 수학과(학사)  
1987년 서울대학교 전산학과(이학석사)  
1995년 서울대학교 전산학과(이학박사)  
1996년~현재 경기대학교 컴퓨터과학과  
교수  
관심분야: 인공지능, 기계학습, 지능형시스템