

# Head Pose Estimation Based on Perspective Projection Using PTZ Camera

Kim Jin Suh<sup>†</sup> · Lee Gyung Ju<sup>\*\*</sup> · Kim Gye Young<sup>\*\*\*</sup>

## ABSTRACT

This paper describes a head pose estimation method using PTZ(Pan-Tilt-Zoom) camera. When the external parameters of a camera is changed by rotation and translation, the estimated face pose for the same head also varies. In this paper, we propose a new method to estimate the head pose independently on varying the parameters of PTZ camera. The proposed method consists of 3 steps: face detection, feature extraction, and pose estimation. For each step, we respectively use MCT(Modified Census Transform) feature, the facial regression tree method, and the POSIT(Pose from Orthography and Scaling with Iteration) algorithm. The existing POSIT algorithm does not consider the rotation of a camera, but this paper improves the POSIT based on perspective projection in order to estimate the head pose robustly even when the external parameters of a camera are changed. Through experiments, we confirmed that RMSE(Root Mean Square Error) of the proposed method improve 0.6° less than the conventional method.

**Keywords :** Head Pose Estimation, Perspective Projection, POSIT, MCT, Regression Tree

## 원근투영법 기반의 PTZ 카메라를 이용한 머리자세 추정

김진서<sup>†</sup> · 이경주<sup>\*\*</sup> · 김계영<sup>\*\*\*</sup>

### 요약

본 논문에서는 PTZ 카메라를 이용한 머리자세추정 방법에 대하여 서술한다. 회전 또는 이동에 의하여 카메라의 외부인자가 변경되면, 추정된 얼굴자세도 변한다. 본 논문에는 PTZ 카메라의 회전과 위치 변화에 독립적으로 머리자세를 추정하는 새로운 방법을 제안한다. 제안하는 방법은 얼굴검출, 특징추출 그리고 자세추정으로 이루어진다. 얼굴검출은 MCT특징을 이용해 검출하고, 얼굴 특징추출은 회귀트리 방법을 이용해 추출하고, 머리자세 추정은 POSIT 알고리즘을 사용한다. 기존의 POSIT 알고리즘은 카메라의 회전을 고려하지 않지만, 카메라의 외부인자 변화에도 강건하게 머리자세를 추정하기 위하여 본 논문은 원근투영법에 기반하여 POSIT를 개선한다. 실험을 통하여 본 논문에서 제안하는 방법이 기존의 방법 보다 RMSE가 약 0.6° 개선되는 것을 확인했다.

**키워드 :** 머리자세 추정, 원근투영법, POSIT, MCT, 회귀트리

### 1. 서론

본 논문에서는 머리자세 추정에 대하여 서술한다. 컴퓨터 비전에서 머리자세 추정이란 이력영상을 자동으로 분석하여 머리의 위치 및 각도를 산출하는 기술을 의미하며, 얼굴인식, 시선추적, 동작인식 등에서 요소기술로 사용되고 있다. 머리자세 추정을 위한 새로운 방법들이 개발됨에 따라 그 정확도

가 향상되고 있다[1]. 하지만, 머리자세 추정을 통해 컴퓨터 시스템과 상호작용[2] 할 수 있는 사용자 인터페이스에 대한 요구가 증대함에 따라 보다 더 정확하고 편리하게 머리자세를 추정하는 방법이 필요하다[3, 4].

머리자세를 추정하는 접근법은 통계적 접근방법, 모델 기반 접근방법, 기하학적 접근방법, 추적 접근방법 등이 있다 [1]. 이들 중에서 모델 기반의 머리자세 추정방법은 영상에서 얼굴 구조와 일치하도록 모델을 정합시켜 머리자세를 산출하는 방법으로, 다른 방법들보다 얼굴 구조에 최대한 정확하게 접근하도록 설계되었기 때문에 다른 방법들보다 정확하게 머리자세를 산출할 수 있어서 최근에는 모델기반 접근법이 각광받고 있다. Seong G. Kong 등은 모핑(morphing)된 3차원

<sup>†</sup> 준회원 : 숭실대학교 컴퓨터학과 석사  
<sup>\*\*</sup> 준회원 : 숭실대학교 컴퓨터학과 박사과정  
<sup>\*\*\*</sup> 종신회원 : 숭실대학교 소프트웨어학부 교수  
Manuscript Received : March 27, 2018  
Accepted : April 28, 2018  
\* Corresponding Author : Kim Gye Young(gykim11@ssu.ac.kr)

얼굴모델을 사용해 얼굴영상에서 머리자세를 추정했다[10]. 모핑된 3차원 얼굴모델은 인종과 성별을 기준으로 구분하였고, 모델의 깊이는 얼굴 특징점들의 매개변수를 조정해 산출했으며 얼굴을 사실적으로 표현하기 위해 모델에 질감정보(texture)를 추가하였다. 얼굴 특징점과 3차원 모델의 대응하는 점들 사이의 차(disparity)를 최소화하는 방법으로 머리자세를 추정했다. 질감정보가 추가된 모델은 얼굴을 구체적으로 표현할 수 있기 때문에 실제 얼굴과 유사한 얼굴모델을 생성할 수 있는 장점이 있다. 하지만 3차원 모델생성시 모델의 깊이점이 단일영상에서 추정되었기 때문에 깊이로 인한 오류가 발생할 문제가 있다. Zhan-Li 등은 얼굴영상에서 비선형 최소자승모델을 이용해 얼굴을 3차원으로 재구성하기 위한 깊이값을 추정하는 알고리즘을 제안했다[11]. 여러 가지 자세를 포함하는 얼굴영상을 3차원으로 재구성하기 위해 첫 번째로 비선형 최소제곱모델을 유사성 변환에 의해 얼굴 특징점의 깊이값과 관련된 머리자세를 추정했다. 머리자세의 변화로 인해 발생하는 민감도를 줄이기 위해 얼굴의 대칭성을 사용한 것이 특징이다. 하지만 모델을 구성하고 최적화시키기 위해 많은 훈련시간이 요구된다. Pedro Martins 등은 단일 카메라 환경에서 AAM(Active Appearance Models)[13]과 POSIT 알고리즘을 이용해 머리자세를 추정하였다[12]. AAM을 통해 실시간으로 얼굴의 특징점을 추출 및 추적하고, POSIT 알고리즘과 3차원 모델을 이용해 머리자세를 추정하였다. 실시간으로 획득한 얼굴 특징점과 3차원 모델을 통해 빠르게 머리자세를 추정할 수 있다. 하지만 3차원 모델과 실제 사람의 얼굴과 유사하지 않기 때문에 발생하는 오차와, 카메라의 위치와 회전변화에 따른 머리자세 추정의 오류가 발생한다. 왜냐하면 영상에서 사용자의 머리자세는 실제 머리자세와 다르게 보이고, 영상에서 보이는 사용자의 머리자세는 원근투영법에 따라 카메라의 회전과 위치변화에 민감하기 때문이다[5]. 실제 사용자의 머리자세의 변화가 없을 때, 카메라의 회전과 위치가 변하면 영상에서 사용자의 머리자세는 변동된다. 따라서 본 논문에서는 카메라의 회전과 위치변화에 독립적으로 머리자세를 추정하는 기법을 제안한다.

실제 사용자의 머리자세 추정을 위해 본 논문에서는 PTZ 카메라를 통해 사용자의 얼굴을 검출하고 원근투영법을 적용해 카메라의 위치와 회전변화에 강건한 머리자세 추정방법을 제안한다. 제안하는 방법의 개요도는 Fig. 1과 같다.

제안하는 방법은 카메라의 광각모드에서 사용자를 감지한 후 얼굴 크기와 위치에 맞춰 줌각모드로 전환한다. 획득한 영상은 카메라의 회전변화에 따라 실제 머리자세와 영상에서의 머리자세가 다르게 보이고, 영상에서 머리 자세를 추정할 경우 오차각이 크게 발생하는 것을 확인할 수 있다. 따라서 원근투영법을 기반으로 카메라의 회전과 위치정보를 통해 3차원 얼굴 모델이 영상에 어떻게 투영될지 예측하고 재구성한다. 얼굴검출 단계에서는 MCT 특징[6]을 이용하고, AdaBoost 분류기[7]를 통해 얼굴을 검출한다. 얼굴의 특징점은 회귀트리

를 이용해 얼굴의 눈, 눈썹, 코끝, 얼굴 윤곽등의 68개 점을 추출한다[8]. 추출된 특징점을 이용해 카메라의 회전과 위치 변화에 따라서 원근투영법을 적용한 개선된 POSIT 알고리즘[9]을 통해 머리 자세를 추정한다.

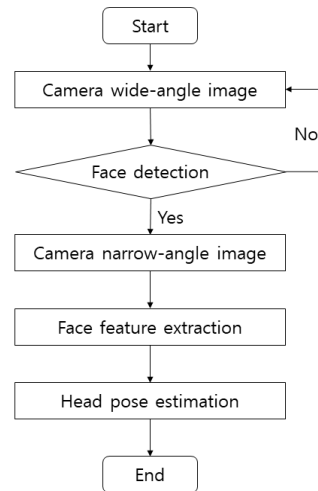


Fig. 1. Proposed System Overview

본 논문은 총 6절로 구성되어 있고, 카메라의 위치와 회전에 독립적으로 머리자세를 추정하기 위해, PTZ 카메라를 이용해 원근투영법 기반의 머리자세 추정방법을 제안한다. 2절에서는 얼굴검출과 카메라 제어방법, 3절에서는 얼굴특징 추출방법, 4절에서는 머리자세 추정방법, 5절에서는 실험과 결과를 분석하고, 6절에서는 결론을 도출한다.

## 2. 카메라 제어 및 MCT 특징을 통한 얼굴검출

이 절에서는 PTZ 카메라 제어방법과 얼굴검출에 관하여 설명한다. PTZ 카메라는 Pan, Tilt, Zoom 기능을 가지고 있는 카메라로 카메라 회전을 통해 얼굴을 추적할 수 있는 장점이 있다[14]. 따라서 고정된 카메라와 달리 화면의 크기나 모양에 구애 받지 않고, 사용자의 위치에 상관없이 카메라의 회전과 확대를 통해 얼굴영상을 획득할 수 있다. 카메라 제어부에서는 카메라의 광각모드, 줌각모드로 전환한다. 즉, 광각 모드에서는 얼굴검출을 통해 사용자의 얼굴을 영상 중앙에 위치시키고, 줌각모드 전환을 위해 확대 배율을 계산한다.

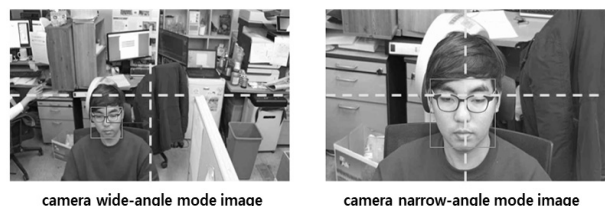


Fig. 2. Camera Mode Example Images

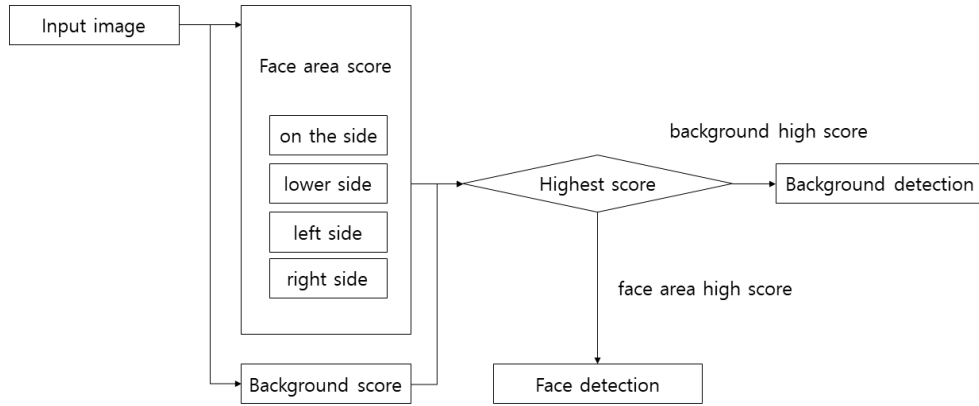


Fig. 3. Face Detection Process Using Face Classifier

Fig. 2는 광각모드에서 협각모드로 전환된 얼굴영상을 보여준다. 협각모드는 광각모드에서 얼굴이 검출되면 카메라의 회전을 통해 영상중심에 얼굴을 위치하고 적절한 크기의 얼굴영상을 획득하기 위해 확대한다. 광각모드에서 얼굴검출은 MCT특징과 AdaBoost 분류기를 이용해 얼굴을 검출한다. MCT특징은 Equation (1)을 통해 변환할 수 있고, 3x3 윈도우에서 0 또는 1을 가지는 지역 구조적 특징이 511개 생성되기 때문에 조명 변화에 강건하고 계산이 간단하여 얼굴검출에 소요되는 계산시간을 단축시킬 수 있다.

$$MCT(x) = \otimes_{y \in dir} X(\bar{I}(x), I(y)) \quad (1)$$

Equation (1)에서  $MCT(x)$ 는 MCT 값을 나타내고,  $x$ 는 윈도우의 중심,  $\bar{I}(x)$ 는 3x3 윈도우 영역의 평균이다.  $dir$ 은 윈도우의 중심과 8방향의 이웃화소를 나타낸다.  $X(\bar{I}(x), I(y))$ 은 비교 함수로  $\bar{I}(x) < I(y)$  이면 1, 그렇지 않으면 0을 반환한다.

얼굴을 검출하기 위해 MCT 특징을 추출하고 AdaBoost 분류기로 얼굴인지 아닌지를 판별한다. 분류기 생성을 위해 얼굴인 영상과 그렇지 않은 영상을 입력으로 한다. 입력영상으로부터 오류율이 낮은 약분류기를 생성하고 약분류기를 통해 하나의 강분류기를 생성한다. 강분류기는 캐스케이드(cascade) 구조로 얼굴검출을 수행한다. 얼굴검출은 정면, 상측면, 하측면, 좌측면, 우측면, 그리고 배경영상으로 구성되어 있다. 분류기에서 얼굴이 검출되었을 때 검출에 대한 점수를 계산하고, 얼굴과 배경의 점수를 비교해 얼굴 혹은 배경을 검출한다.

검출된 얼굴로부터 광각모드에서 협각모드로 변환한다. Fig. 3은 얼굴분류기를 이용한 얼굴검출 과정을 보여준다. 정면을 포함한 각 측면의 얼굴영역에 대한 점수와 배경에 대한 점수가 각각 계산되고, 배경과 얼굴영역 중 높은 점수를 획득한 부분이 검출된다. Fig. 4는 협각모드에서 검출된 얼굴의 결과영상이다. 정면을 포함한 각 측면의 얼굴까지 검출이 잘 된 것을 확인할 수 있다.



Fig. 4. Face Detection Result Images

### 3. 회귀트리를 통한 얼굴특징 추출

얼굴특징 추출은 협각모드에서 진행되며, 머리자세를 추정하기 위해 형상 기반의 얼굴특징 추출 방법을 사용한다. 형상 기반의 얼굴특징 추출 방법들 중에 ASM(Active Shape Model)과 AAM가 대표적인 이다. 형상을 구성해 입력영상에서 형상이 매개변수에 의해 변화하며 객체를 검출한다. 실시간 시스템에서 AAM보다 ASM의 속도가 더 빠르지만 정확도는 AAM이 높다. 하지만 회귀트리를 이용한 형상 기반의 얼굴특징은 실시간 시스템에 적합하고 ASM보다 높은 정확도를 보인다. 따라서 본 논문에서는 Equation (2)를 통해 회귀트리를 이용하여 얼굴특징을 추출한다. 얼굴특징은 얼굴에서 의미 있는 눈, 코, 입 등을 포함한 68개의 점을 추출한다.

$$S^t = S^{t-1} + r^t(I_f, S^{t-1}) \quad (2)$$

Equation (2)에서  $S^t$ 는 68개의 점을 가진 최종 형상이고,  $r^t$ 는  $t$ 번째 약한 복귀자(weak regressor)를 표현한다. 약한 복귀자는 형태 증가분을 계산하고, 얼굴 형상을 변형해 얼굴의 특징위치에 대응되는 특징점을 추출한다[8].

Fig. 5는 회귀트리를 이용해 얼굴의 특징점을 추출한 예이다. 초록색 사각형은 얼굴검출 영역을 표현하고 68개의 빨간 점이 얼굴의 특징점이다. 추출된 얼굴의 형상과 모델을 통해

머리자세 추정을 한다. 머리자세의 추정을 위해 사용되는 모델은 얼굴특징의 68개점과 대응되는 68개의 점을 포함하고 POSIT 알고리즘을 이용해 머리자세를 추정한다.

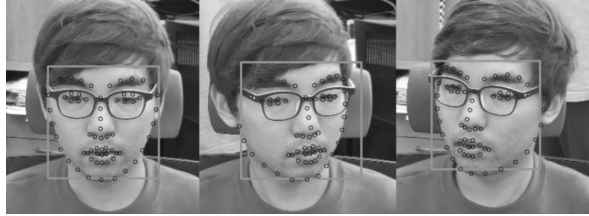


Fig. 5. Face Features Extraction Result Images

#### 4. 머리자세 추정을 위한 POSIT 개선

본 논문에서는 얼굴의 특징점과 3차원 모델의 좌표들을 이용해 머리자세를 추정하기 위해 POSIT 알고리즘을 이용한다. 하지만 기존의 POSIT 알고리즘은 고정 카메라 환경에 최적화 되어있어 사용자 위치에 대한 제약이 있고, 카메라의 초점거리만을 이용하는 방법이기 때문에 카메라의 위치와 회전변화에 따라 추정되는 결과가 민감하다. 카메라의 자세변화에 대한 정보를 이용해 POSIT 알고리즘에 적용해 카메라의 위치와 회전변화에 강건한 머리자세 추정방법을 제안한다. Fig. 6은 각각의 축의 회전에 대한 머리자세를 정의한다. X, Y, Z축 회전을 Tilt, Pan, Swing 각도로 표현한다. 최종적으로 머리자세 추정 값은 Pan, Tilt, Swing 각도로 표현한다.

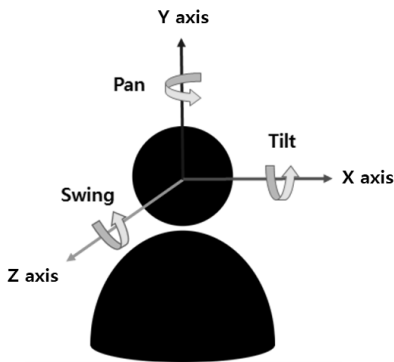


Fig. 6. Face Rotation Angle Definition

POSIT 알고리즘은 “POS(Pose from Orthography and Scaling)”과 “with Iteration”이 결합된 알고리즘이다. 약한 원근투영근사화(weak-perspective approximation)를 가정하고 있다. 이것은 3차원 객체가 카메라와 멀리 떨어져 있기 때문에 모든 점이 동일한 깊이를 가진다는 의미다. 동일한 깊이를 가지게 되면 3차원 모델에 회전행렬의 곱으로 영상에서 추출된 특징점 좌표를 Equation (3)으로 표현할 수 있다. Fig. 7의 핀홀 카메라 모델에서 각각의 축의 단위벡터들로 이루어

진 회전행렬을 구할 수 있다. Equation (3)의 회전행렬에서  $i_u, i_v, i_w$ 는 객체의 좌표 체계( $M_0u, M_0v, M_0w$ )에서  $i$ 의 좌표이다.  $k$ 는  $i$ 와  $j$ 의 외적으로 계산할 수 있다.

$$R = \begin{bmatrix} i_u & i_v & i_w \\ j_u & j_v & j_w \\ k_u & k_v & k_w \end{bmatrix} \quad (3)$$

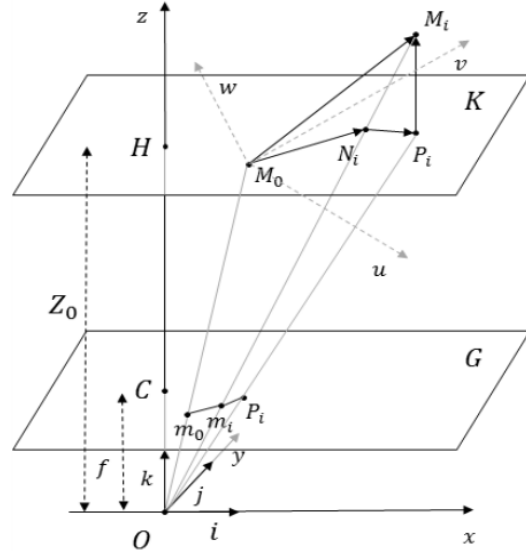


Fig. 7. Pin Hole Camera Model

Equation (4)은 이동벡터를 구하는 식이다.  $Z_0$ 은 카메라 투영의 중심과 객체의 참조점 사이의 거리이다.  $f$ 는 카메라의 초점거리,  $O_{m_0}$ 은 카메라 투영의 중심으로부터 영상평면에 투영된 참조점과의 거리이다. 회전행렬과 이동벡터를 3차원 모델에 적용하여 머리자세를 추정할 수 있다. 3차원 모델이 영상평면에 투영될 때, 오직 크기 요소만을 고려해 투영된다.  $i$ 번째의 특징점은  $M_0M_i$ 을 통해 계산한다.

$$T = \frac{Z_0}{f} O_{m_0} \quad (4)$$

$K$  평면의 3차원 모델 좌표  $M_i$ 는 영상평면에  $m_i$ 에 투영된다. Equation (5)를 통해서  $m_i$ 를 구할 수 있다.  $s$ 는 크기 요소로 초점거리와 카메라 투영의 중심과 3차원 모델의 참조점 사이의 거리 비율로 구한다. 3차원 모델의 투영된 점과 얼굴의 특징점과 같다고 가정하고 계산할 수 있다.

$$\begin{aligned} x'_i &= x_0 + s(X_i - X_0) \\ y'_i &= y_0 + s(Y_i - Y_0) \end{aligned} \quad (5)$$

Equation (6)에서  $\epsilon_i$ 는  $\frac{1}{Z_0} M_0M_i \cdot k$ 로 정의되고  $k$ 는  $i \times j$ 로



정의된다. 회전 행렬의 첫 번째 행은  $M_0u, M_0v, M_0w$ 의  $i$ 번째 좌표이다.  $\epsilon_i$ 을 구하기 위해 초기값은 0으로 설정하고 증가시키면서 반복수행한다. 임계치를 설정해 임계치보다 높으면 반복을 중지하고 계산된  $\epsilon_i$ 값을 통해  $\frac{f}{Z_0}i$ 와  $\frac{f}{Z_0}j$ 을 계산한다.

$$M_0M_i \cdot \frac{f}{Z_0}i = x_i(1+\epsilon_i) - x_0 \quad (6)$$

$$M_0M_i \cdot \frac{f}{Z_0}j = y_i(1+\epsilon_i) - y_0$$

Equation (7)에서  $A$ 는 3차원 모델의 좌표이고,  $I = \frac{f}{Z_0}i$ 와  $J = \frac{f}{Z_0}j$ 로 정의한다.  $x'$ 은  $A$ 와  $I$ 의 곱으로 특징점의 좌표,  $y'$ 은  $A$ 와  $J$ 의 곱으로 특징점의 좌표를 나타낸다.  $I$ 와  $J$ 의 단위벡터를 계산하기 위해 최소자승법(Least Square Method) [15]을 사용한다.

$$AI = x', AJ = y' \quad (7)$$

Equation (8)에서  $B$ 는  $A$ 의 의사역행렬이다.  $B$ 를 이용해  $I$ 와  $J$ 를 구할 수 있다.  $I$ 와  $J$ 를 이용해  $i, j, k$ 를 계산 할 수 있다. 최종적으로 회전행렬과 이동벡터를 산출해 사용자의 머리자세를 추정할 수 있다. 하지만 카메라의 위치와 회전에 따라서 원근투영법에 의해 영상에 보이는 사용자는 실제 자세와는 다르게 보인다. 따라서 원근투영법[16]을 기반으로 카메라의 회전정보와 위치를 통해 3차원 모델이 영상에서 어떤 자세로 투영될지 예측하고 재구성한다. 이 정보들을 기반으로 최종적으로 3차원 모델을 변환한다. 재구성된 3차원 모델은 카메라의 위치와 회전변화가 발생했을 때, 원근법을 기반으로 기준자세로 변환한다. 기준자세는 카메라의 위치와 회전 중심으로 카메라와 정면을 응시하는 자세인  $pan, tilt, swing$ 이 각각  $0^\circ$ 이다. Equation (9)에서  $CP_r$ 은 회전행렬이다. 카메라 회전정보와 위치정보를 포함하고 있다.  $\alpha$ 는  $pan$ ,  $\beta$ 는  $tilt$ 를 표현하고 있다. Equation (10)을 이용해 3차원 모델을 카메라 회전변화와 위치에 따라 기준자세로 변환한다.  $M = [x_m, y_m, z_m]^T$ 은 변환되지 않은 3차원 모델의 좌표이고,  $C = [x_c, y_c, z_c]^T$ 는 카메라 캘리브레이션을 통해 산출한 카메라의 위치이다. 최종적으로 산출된  $3DPoint = [x, y, z]^T$ 는 변환된 모델의 좌표이다.

$$I = Bx', J = By'$$

$$B = (A^T A)^{-1} A^T \quad (8)$$

$$CP_r = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ \sin\alpha\sin\beta & \cos\beta & \cos\alpha\sin\beta \\ \sin\alpha\cos\beta & -\sin\beta & \cos\alpha\cos\beta \end{bmatrix} \quad (9)$$

$$3DPoint = CP_r(C - M) \quad (10)$$

$$3DPoint_x I = x'$$

$$3DPoint_y J = y' \quad (11)$$

Equation (10)을 통해 변환된  $3DPoint$ 를 이용해 Equation (7)을 변형시켜 머리자세를 추정할 수 있는 수식을 만들 수 있다. Equation (11)은 기준자세의 모델과 특징점  $x', y'$ 와 단위벡터  $I, J$ 의 관계를 나타내고 있다. 따라서  $3DPoint$ 의 의사역행렬을 구해  $I, J$ 를 산출해낼 수 있고, 최종적으로  $i, j, k$ 를 계산해 사용자의 머리자세를 추정할 수 있다.

### 5. 실험 및 결과

원근법 기반의 머리자세 추정 방법을 실험하고 성능 평가를 한다. 성능 평가는 기존의 머리자세 추정 알고리즘인 POSIT 알고리즘과 제안한 방법으로 추정한 머리자세 추정 각도의 오차를 비교한다. 실험에는 카메라의 회전 변화를 포함하고 있다.

실험을 위해서 사용한 컴퓨터는 Intel Core i7-2600 3.40GHz, RAM 32.0GB를 사용하였고, 운영체제는 Windows 8.1 Pro, 개발 언어는 C/C++로 Visual Studio 2010 32비트와 라이브러리는 dlib, Opencv 2.4.10을 사용했다. 실험을 위해 GT(Ground Truth)를 제작하였는데, GT는 E2BOX의 EBIMU-9DOF3 센서를 이용하였다. 센서를 통해 실험할 사용자의 머리자세를 산출했다. 생성된 GT를 통해 POSIT 알고리즘과 제안한 방법을 통해 머리자세 각도를 추정하고 비교하였다. 카메라의 위치는 지면으로부터 약 170cm 위치에 설치해 디스플레이 상단에 카메라가 있다고 가정하였다. 카메라의 회전 범위는 Pan각은  $-15^\circ \sim 15^\circ$ , Tilt각은  $-25^\circ \sim -10^\circ$ 이다. GT는 총 879프레임으로 Pan각도 기준으로 146-137-141-145-157-153 프레임으로 구성했다.

Fig. 8은 GT의 일부분을 나타낸 것이다. 카메라의 회전을 기준으로 사용자의 머리자세 각도가 포함되어 있다. POSIT 알고리즘으로 머리자세 추정을 했을 때 Tilt각이 최대  $20^\circ$ 까지 오차가 발생하는 것을 확인할 수 있었다. 하지만 제안하는 방법으로 머리자세 추정을 했을 때 Tilt각이 최대  $7^\circ$ 까지 발생하는 것을 확인할 수 있었다. 머리자세 추정에 대한 성능 평가는 RMSE(Root Mean Square Error)로 평가했다. RMSE는 제안한 방법을 통해 산출된 머리자세 각도가 GT에 근접하는 정도를 평균 제곱근 오차로 나타낸 것이다. Table 1은 RMSE 측정 결과이다. Tilt각에서 차이가 많이 발생하는 것을 확인할 수 있다. 이것은 카메라를 상단에 설치하고, Tilt각이 최대  $-25^\circ$ 까지 변화하기 때문에 나타난 결과이다.

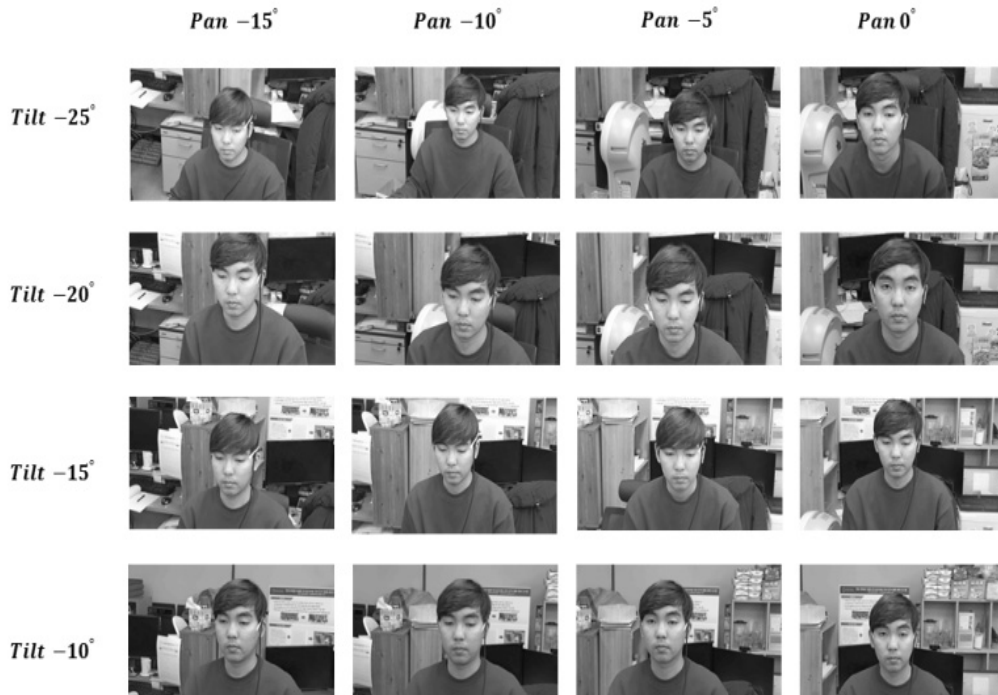


Fig. 8. Head Pose Ground Truth Images

Table 1. RMSE Results of Posit Algorithm and Proposed Method

RMSE	Pan	Tilt	Swing
Posit algorithm	0.230834	0.861226	0.165985
Proposed	0.108778	0.222567	0.117751

Figs. 9-11은 일정 프레임구간에서의 GT, 기존 POSIT 알고리즘, 제안한 방법의 머리자세 각도 추정 결과를 그래프로 나타내었다. Pan과 Swing의 각도의 경우 POSIT과 비교한 결과 약간의 성능개선이 된 것을 확인할 수 있다. Tilt각도의 경우에는 카메라의 회전과 위치변화에 독립적으로 머리자세를 추정하기 때문에 획기적으로 각도 추정 오차범위가 줄어든 것을 볼 수 있다.

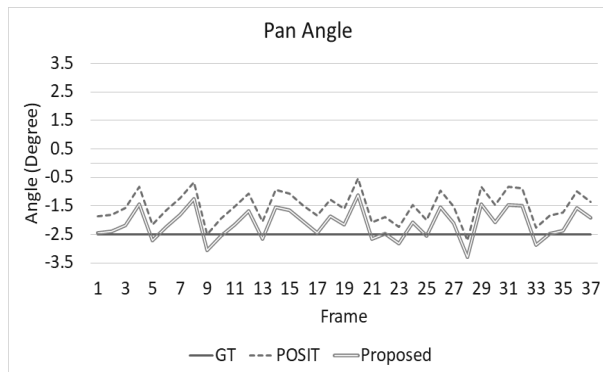


Fig. 9. Pan Angle Comparison by Method

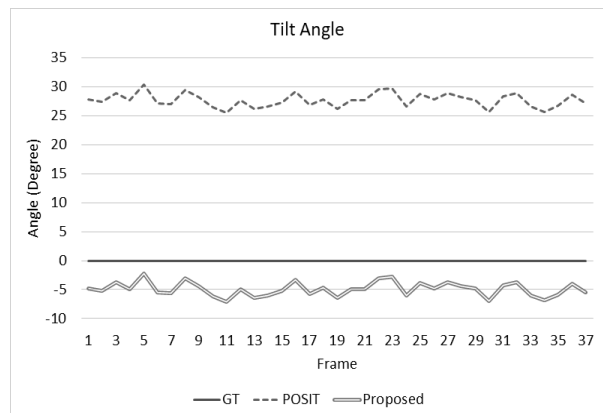


Fig. 10. Tilt Angle Comparison by Method

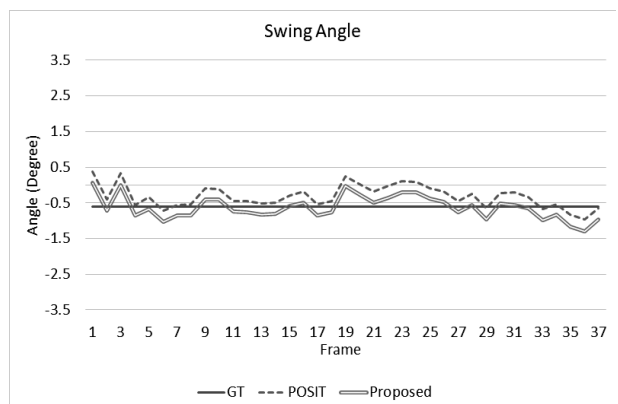


Fig. 11. Swing Angle Comparison by Method

## 6. 결 론

머리자세 추정은 얼굴 인식, 시선 추적 등 많은 분야에 중요한 요소기술로 적용되고 있다. 하지만 기존의 연구들은 다양한 환경에 따라 제약적이며 대부분 모니터를 이용한 환경이 많다. 하지만 본 논문에서는 PTZ 카메라를 사용하여 환경에 제약적인 사항을 완화시키고, 사용자가 카메라 중심에 위치하지 않아도 카메라를 회전시켜 사용자의 머리자세를 추정하는 방법을 제안하고 실험을 통하여 그 성능을 확인한 결과, 카메라의 회전이 발생했을 경우 기존의 머리자세 추정방법보다 좋은 성능을 보였다. 향후 머리자세 추정 시스템이 다양한 환경 변화에서 강건하게 머리자세를 추정할 수 있도록 연구된다면, 카메라의 회전뿐만 아니라 다른 환경에서도 머리자세 추정을 할 수 있을 것으로 생각한다.

## References

[1] B. Czupryński and A. Strupczewski, "High Accuracy Head Pose Tracking Survey," *Proc. International Conference on Active Media Technology*, pp.407-420, 2014.

[2] Z., Feng, H. B.-L. Duh, and M. Billinghurst, "Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR," *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Cambridge, UK. (2008 September).  
<http://dx.doi.org/10.1109/ismar.2008.4637362>.

[3] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," *IEEE Conference on Computer Vision and Pattern Recognition*, pp.617-624, 2011.

[4] C. Huang, X. Ding, and C. Fang, "Head pose estimation based on random forests for multiclass classification," *International Conference on Pattern Recognition*, pp.934-937, 2010.

[5] Y. Ohta, K. Maenobu, and T. Sakai, "Obtaining Surface Orientation from Texels Under Perspective Projection," *Proc. Seventh Int'l Joint Conf. Artificial Intelligence*, pp.746-751, 1981.

[6] R. Zabih and J. Woodfill, "Non-parametric Local Transforms for Computing Visual Correspondence," pp.151-158, 1994.

[7] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, pp.119-139, 1997.

[8] V. Kazemi and J. Sullivan, "One Millisecond Face Alignment with an Ensemble of Regression Trees," *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867-1874, 2014.

[9] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, Vol.15, No.1-2, pp.123-141, 1995.

[10] S. G. Kong and R. O. Mbouna, "Head Pose Estimation From a 2D Face Image Using 3D Face Morphing With Depth Parameters," *IEEE Transactions on Image Processing*, Vol.24, No.6, pp.1801-1808, 2015.

[11] Z. L. Sun, K. M. Lam, and Q. W. Gao, "Depth Estimation of Face Images Using the Nonlinear Least-Squares Model," *IEEE Transactions on Image Processing*, pp.17-30, 2013.

[12] P. Martins and J. Batista, "Single view head pose estimation," *IEEE International Conference on Image Processing*, pp. 1652-1655, 2008.

[13] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.681-685, 2001.

[14] T. Funahashi, M. Tominaga, T. Fujiwara, and H. Koshimizu, "Hierarchical face tracking by using PTZ camera," *IEEE International Conference on Automatic Face and Gesture Recognition*, pp.427-432, 2004.

[15] S. J. Miller, "The Method of Least Squares," in Brown University, 2006.

[16] W. J. Wolfe, D. Mathis, C. W. Sklair, and M. Magee, "The perspective view of three points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.66-73, 1991.



김진서

<https://orcid.org/0000-0002-5869-1129>

e-mail : kimjinsuh@ssu.ac.kr

2003년 서울호서전문학교 사이버해킹  
보안과, 국가평생교육진흥원  
컴퓨터공학(학사)

2017년 숭실대학교 컴퓨터학과(석사)

관심분야 : Pattern Recognition & Computer Vision



이경주

<https://orcid.org/0000-0002-1602-3114>

e-mail : lkj0917@ssu.ac.kr

2010년 군산대학교 컴퓨터정보공학과(학사)  
2013년 숭실대학교 컴퓨터학과(석사)  
2013년~현 재 숭실대학교 컴퓨터학과  
박사과정

관심분야 : Pattern Recognition & Computer Vision



## 김 계 영

<https://orcid.org/0000-0001-6908-6920>

e-mail : [gykim11@ssu.ac.kr](mailto:gykim11@ssu.ac.kr)

1990년 숭실대학교 전자계산학과(학사)

1992년 숭실대학교 컴퓨터학과(석사)

1996년 숭실대학교 컴퓨터학과(박사)

현 재 숭실대학교 소프트웨어학부 교수

관심분야: Computer Vision, Pattern Recognition, Augmented Reality