

# Metaverse 영상 제작에 사용되는 카메라들의 파라미터 최적화를 위한 Bundle Adjustment

□ 최건찬, 한종기 / 세종대학교

## 요약

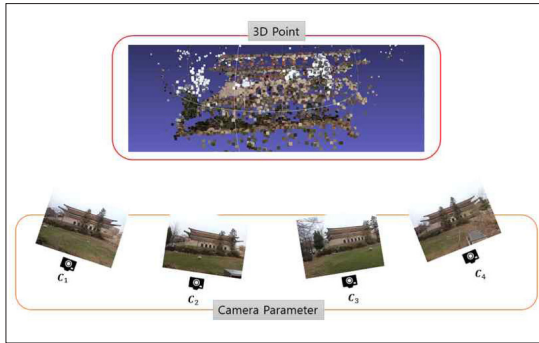
360VR이나 SFM(Structure From Motion)과 같은 기술들은 Metaverse 환경을 구성하기 위해 사용될 수 있다. Bundle Adjustment는 이 기술들의 성능을 높이기 위한 최적화 방법으로, Bundle Adjustment를 수행하지 않으면 에러가 누적되어 좋은 품질의 결과 영상을 얻기 어렵다. 본 글에서는 Bundle Adjustment의 개념 및 자세한 알고리즘을 설명한다.

## 1. 서론

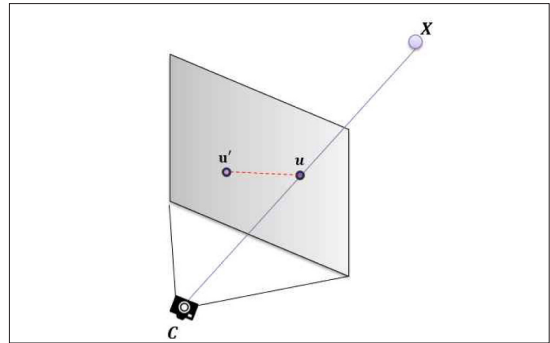
Metaverse 또는 VR과 같은 기술들은, 영상을 이용한 콘텐츠 및 기술의 확장을 가능하게 하는 기술들 중 하나로써 주목받고 있다. 이를 구현하기 위해서는 영상 합성 기술이나 입체 영상 제작 기술이 사용된다. 따라서 360VR, SFM(Structure From Motion)[1], SLAM(Simultaneous Localization and Mapping)[2]과 같은 분야에 대한 연구들이 이뤄지고 있으며, 이러한 작업들에서 Bundle Adjustment를 수행함으로써 error가 적고 품질이 좋

은 결과 영상을 얻을 수 있다. 결과 영상을 얻기 위해서는 camera parameter와 3D point를 올바르게 추정하는 것이 중요한데, 이를 최적화하는 작업을 Bundle Adjustment라고 한다. 본 글에서는 Bundle Adjustment의 개념 및 알고리즘에 대하여 설명한다.

3D point cloud를 제작하는 기술 중 하나인 SFM[1]의 경우, 각 이미지에서 특징점들을 추출하고 특징점간의 matching 관계를 기반으로 각 이미지 쌍마다 camera parameter와 3D point 좌표를 추정한다. 이렇게 추정한 camera parameter와 3D point 좌표는 어느 정도의



&lt;그림 1&gt; Bundle Adjustment의 개념



&lt;그림 2&gt; Reprojection Error

오차를 가질 수밖에 없는데, 다수의 이미지들을 모두 결합하는 과정에서 오차값이 점점 커지게 된다. Bundle Adjustment는 camera parameter들과 3D point들을 최적화할 파라미터로 취급하고, 이를 조금씩 조정하여 오차값을 줄이는 방법이다. 매 시행마다 파라미터들을 update 시킴으로써 오차값을 점점 줄여나가고, 최종적으로 얻은 파라미터들을 이용하여 입체 영상을 제작한다.

Bundle Adjustment 작업을 통해 결과 영상의 품질을 개선시킬 수는 있지만, 사용되는 이미지의 개수가 증가할수록 Bundle Adjustment 작업에서의 계산량이 상당히 커진다는 문제점이 있다. Bundle Adjustment 작업의 수행 시간이 굉장히 오래 걸려서 전체 결과 영상 생성 시간이 길어지는데, 이를 개선하기 위한 연구들도 발표되고 있다.

## II. Camera Parameter 및 3D point 최적화 문제 설명

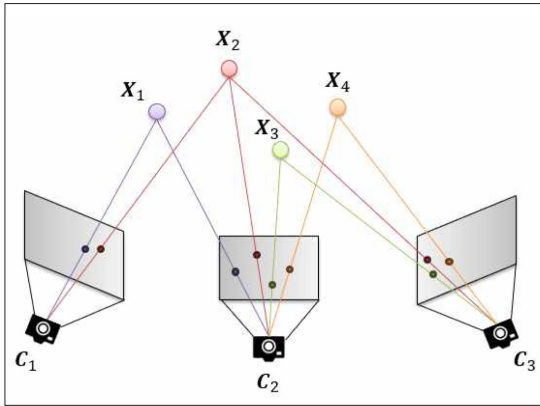
이미지 또는 특징점간의 matching 관계를 이용하여 초기 camera parameter와 초기 3D point들을 추정할 수 있다. 하지만 이렇게 추정된 초기 파라미터들은 제법 큰 reprojection error를 가진다. <그림 2>를 보면, ‘추정된 3D point인  $X$ 를 이미지로 reprojection시킨 point  $u'$ 와

‘관찰된 특징점  $u$ ’ 사이에 error가 존재하는 것을 확인할 수 있다. 이  $u-u'$ 을 reprojection error라고 한다. 결과 영상 제작에 사용되는 사진과 3D point의 수가 많아질수록 이 reprojection error가 점점 누적되고, error가 크게 발생하므로 당연히 결과 영상의 품질이 떨어지게 된다. 이 error가 작아지도록 최적화 알고리즘을 적용하는 방법이 Bundle Adjustment이다.

<그림 3>과 같이 3D point cloud에는 다수의 3D point가 포함되어 있고 여러 개의 사진이 사용된다. 관찰된 특징점의 위치는 고정되어 있으므로 reprojection error는 각 camera parameter와 각 3D point의 좌표를 이용하여 계산할 수 있다. 따라서 ‘camera parameter’와 ‘3D point 좌표’를 최적화할 파라미터로 둔다. 실제 사용할 수 있는 결과 영상을 만들기 위해서는 많은 사진을 사용해야 하므로 당연히 reprojection error의 개수가 많아지는데, 이 ‘reprojection error 크기의 총합’을 cost function로서 사용한다. Cost function을  $E$ 라고 하면 식 (1)로 정의할 수 있다.

$$E = \frac{1}{2} \sum_{k=1}^n \|u_k - u'_k\|^2 \quad (1)$$

이 cost function을 최소화하는 것이 Bundle Adjustment의 목적이고 각 error들은 camera parameter 및 3D point 좌표에 대하여 비선형 변수이므로, 이를 비선형 최



<그림 3> 최적화시킬 파라미터

소자승 문제로 볼 수 있다.

### III. Bundle Adjustment

이번 장에서는 Bundle Adjustment에서 사용되는 최적화 알고리즘인 Gradient Descent, 가우스-뉴턴법, Levenberg-Marquardt[3]에 대하여 자세히 설명한다. 본문 글에서는 분모 중심 표현을 사용한다.

편의상 최적화시키고자 하는 모든 각  $i$ 번째 파라미터를  $x_i$ 라고 하고 파라미터의 총 개수를  $n$ 개라고 한다. 또한 식 (2)와 같이 이  $x_i$ 들을 모아놓은 벡터를  $\mathbf{P}$ 라고 한다.

$$\mathbf{P} = [x_1 \ x_2 \ \dots \ x_n]^T \quad (2)$$

마찬가지로 각 error값을  $f_i$ 라고 하고, error값의 총 개수를  $m$ 개라고 한다. 그리고 error를 모아 놓은 벡터를  $\mathbf{F}$ 라고 하면, 식 (3)과 같다.

$$\mathbf{F} = [f_1 \ f_2 \ \dots \ f_m]^T \quad (3)$$

Cost function  $E(\mathbf{P})$ 는 ‘error값 제곱의 총합’으로, 식 (4)와 같다.

$$E(\mathbf{P}) = \frac{1}{2} \sum_{i=1}^m f_i(\mathbf{P})^2 = \frac{1}{2} \mathbf{F}(\mathbf{P})^T \mathbf{F}(\mathbf{P}) \quad (4)$$

#### 1. Gradient Descent

Gradient Descent는 cost function의 극소점을 찾기 위해 cost function의 gradient를 사용하는 방법이다.  $E(\mathbf{P})$ 의 gradient  $\nabla E(\mathbf{P})$ 는  $E(\mathbf{P})$ 에서 가장 급격하게 증가하는 방향을 가리키는데, 따라서  $\nabla E(\mathbf{P})$ 와 반대방향으로 파라미터를 update하여 cost를 감소시킬 수 있다. 이를 식으로 표현하면 식 (5)와 같이 나타낼 수 있다.  $k$ 번 update된 파라미터 벡터를  $\mathbf{P}_k$ 라고 할 때,  $\mathbf{P}_k$ 에서  $\nabla E(\mathbf{P})$ 의 반대방향으로 이동시킨 벡터를 다음 파라미터  $\mathbf{P}_{k+1}$ 로 지정한다.

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mu_k \nabla E(\mathbf{P}) \quad (5)$$

이때 식 (4)에서  $E(\mathbf{P}) = \frac{1}{2} \mathbf{F}(\mathbf{P})^T \mathbf{F}(\mathbf{P})$  이므로, 식 (6)과 같이 치환할 수 있다.

$$\nabla E(\mathbf{P}) = \frac{\partial E(\mathbf{P})}{\partial \mathbf{P}} = \frac{\partial \mathbf{F}(\mathbf{P})}{\partial \mathbf{P}} \mathbf{F}(\mathbf{P}) = \mathbf{J}_F(\mathbf{P})^T \mathbf{F}(\mathbf{P}) \quad (6)$$

식 (5)에서 식 (6)을 대입하면 식 (7)이 된다.

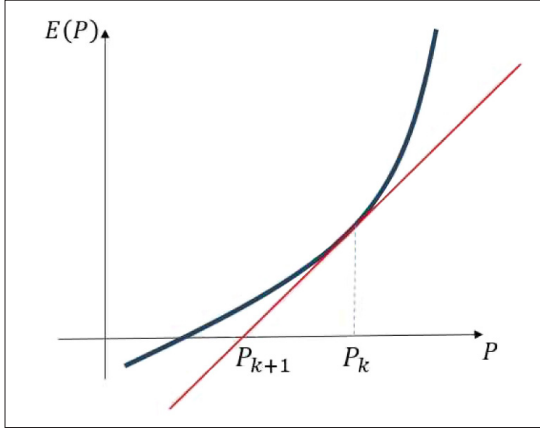
$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mu_k \mathbf{J}_F(\mathbf{P}_k)^T \mathbf{F}(\mathbf{P}_k) \quad (7)$$

수식 (7)에서  $\mu_k$ 는 step size를 의미한다. 즉,  $\mu_k$ 를 조절함으로써 각 iteration에서 파라미터  $\mathbf{P}$ 를 얼마나 update할지 결정할 수 있다. Gradient Descent는 안정적으로 극솟값을 잘 찾을 수는 있지만, 기울기의 크기에 따라 수렴 속도가 영향을 받는다. 극소점 근처에서는 경사가 낮은 경우가 많으므로, 극소점 근처에서 수렴 속도가 느려질 수 있다는 문제점이 있다.

#### 2. 가우스-뉴턴법

가우스-뉴턴법은 뉴턴법을 변형한 방법이므로 먼저 뉴

턴법에 대하여 설명한다. 뉴턴법은 cost function이 0이 되는 파라미터  $P$ 를 찾기 위해, 접선의 방정식이 0이 되는 지점으로 이동하는 방법이다.



<그림 4> 뉴턴법

<그림 4>에서 뉴턴법 알고리즘을 나타내고 있다. 편미 상 파라미터  $P$ 가 1차원일 때(이 1차원 파라미터를  $P$ 로 표기)라고 생각한다. 먼저, 현재 파라미터  $P_k$ 에서  $E(P)$ 의 접선(<그림 4>의 빨간선)을 구한다. 그리고 그 접선이  $P$ 축과 만나는 지점을 다음 파라미터  $P_{k+1}$ 로 정한다. 이 과정을 반복하여 cost function이 0이 되는 파라미터를 찾는다. 이를 식으로 나타내면 수식 (8)과 같다.

$$P_{k+1} = P_k - \frac{E(P_k)}{E'(P_k)} \quad (8)$$

위의 방법은 cost function이 0이 되는 지점을 찾는 방법이다. 하지만 실제로는 cost function이 0이 될 수 없는 경우가 많으므로, 'cost function이 0이 되는 파라미터  $P$ '가 아니라 'cost function이 최소가 되는 파라미터  $P$ '를 찾아야 한다. 따라서,  $E(P) = 0$ 이 되는 지점이 아니라  $E'(P) = 0$ 이 되는 지점을 찾는다. 이를 식으로 나타내면 수식 (9)와 같다.

$$P_{k+1} = P_k - \frac{E'(P_k)}{E''(P_k)} \quad (9)$$

이제 다시 식 (9)의 파라미터  $P$ 를 다차원 벡터  $P$ 로 확장하여 뉴턴법을 수식으로 나타내면 식 (10)과 같다.

$$P_{k+1} = P_k - H_E(P_k)^{-1} J_F(P_k)^T F(P_k) \quad (10)$$

즉, 뉴턴법에 따르면 2차 미분을 통해 Hessian 행렬을 구해야 하므로 계산량이 많다는 단점이 있다.

가우스-뉴턴법은 각 error  $f_i$ 를 1차 함수로 선형 근사하여 뉴턴법을 적용한 방법이다.  $f_i(P)$ 를 1차식으로 근사시키면  $H_E \approx J_F(P)^T J_F(P)$ 로 근사시킬 수 있는데, Jacobian 행렬만 계산하고 직접 2차 미분으로 Hessian 행렬을 구하지 않아도 되므로 뉴턴법에 비해서 필요한 계산량이 적어진다는 장점이 있다. 뉴턴법 수식 (10)에  $H_E \approx J_F(P)^T J_F(P)$ 를 적용한 가우스-뉴턴법은 수식 (11)을 따른다.

$$P_{k+1} = P_k - \{J_F(P_k)^T J_F(P_k)\}^{-1} J_F(P_k)^T F(P_k) \quad (11)$$

수식 (11)을 보면  $J_F(P_k)^T J_F(P_k)$ 의 역행렬을 계산하게 되는데,  $J_F(P_k)^T J_F(P_k)$ 의 역행렬이 존재하지 않을 경우의 사역행렬을 사용해야 하므로 상대적으로 불안정한 값을 가질 수도 있다는 단점이 있다.

### 3. Levenberg-Marquardt[3]

Levenberg-Marquardt 방법은 경사하강법과 가우스-뉴턴법의 원리가 결합된 방법으로, Levenberg 방법[4]의 개선된 형태이다. 먼저, Levenberg 방법의 수식은 수식 (12)와 같다.

$$P_{k+1} = P_k - [J_F(P_k)^T J_F(P_k) + \lambda_k I]^{-1} J_F(P_k)^T F(P_k) \quad (12)$$

수식 (12)의 damping parameter  $\lambda_k$ 의 값이 매우 높다고 가정하면, 수식 (13)과 같이 근사할 수 있을 것이다.

$$\begin{aligned} P_{k+1} &\approx P_k - [\lambda_k I]^{-1} J_F(P_k)^T F(P_k) \\ &= P_k - \frac{1}{\lambda_k} J_F(P_k)^T F(P_k) \end{aligned} \quad (13)$$

수식 (13)을 보면, damping parameter  $\lambda_k$ 가 커질수록 Gradient Descent 수식과 가까워진다는 것을 확인할 수 있다. 반대로,  $\lambda_k$ 의 값이 매우 작다고 가정하면 가우스-뉴턴법 수식인 수식 (11)로 근사될 수 있다는 것을 확인할 수 있다. 즉, Levenberg 방법은 Gradient Descent의 특성과 가우스-뉴턴법의 특성을 모두 지니게 된다. 또한,  $\lambda_k$ 를 조절하면서 두 방법 중 어느 방법에 가깝게 동작할지 조절할 수 있다. 예를 들어, 수렴 속도를 증가시키고 싶을 때는 가우스-뉴턴법처럼 동작하도록  $\lambda_k$ 를 작은 값으로 설정할 수 있다. 또한 update를 해도 오히려 cost function이 증가하고 있으면, 더 안정적으로 최적화를 진행시키기 위해  $\lambda_k$ 를 크게 설정하여 Gradient Descent처럼 동작하도록 할 수 있다. Levenberg-Marquardt 방법[3]도 Levenberg 방법[4]과 유사한데, Levenberg-Marquardt 방법의 수식은 수식 (14)와 같다.

$$\begin{aligned} P_{k+1} &= P_k - [J_F(P_k)^T J_F(P_k) + \\ &\quad \lambda_k \text{diag}\{J_F(P_k)^T J_F(P_k)\}]^{-1} J_F(P_k)^T F(P_k) \end{aligned} \quad (14)$$

수식 (14)를 보면 수식 (13)에서 단위행렬  $I$ 가  $\text{diag}\{J_F(P_k)^T J_F(P_k)\}$ 로 바뀐 것을 확인할 수 있다. 앞서 가우스-뉴턴법에서  $H_E \approx J_F(P)^T J_F(P)$ 로 근사하였으므로,  $\text{diag}\{J_F(P_k)^T J_F(P_k)\}$ 는 Hessian 행렬의 주대각 원소들을 취한 대각 행렬을 의미한다. Hessian 행렬의 원소들은 2차 미분값을 가지므로  $E(P)$ 의 곡률을 반영하는데, 따라서 Gradient Descent의 비중이 높을 때에도 곡률값을 사용하여 최적화를 진행할 수 있다.

가우스-뉴턴법이나 Levenberg-Marquardt 방법을 보

$$J_F(P_k)^T J_F(P_k) = \begin{bmatrix} \frac{\partial F(P)}{\partial A_c} \\ \frac{\partial F(P)}{\partial A_p} \end{bmatrix} \begin{bmatrix} \frac{\partial F(P)}{\partial A_c}^T & \frac{\partial F(P)}{\partial A_p}^T \end{bmatrix} = \begin{bmatrix} \frac{\partial F(P)}{\partial A_c} & \frac{\partial F(P)}{\partial A_c}^T & \frac{\partial F(P)}{\partial A_c} & \frac{\partial F(P)}{\partial A_p}^T \\ \frac{\partial F(P)}{\partial A_p} & \frac{\partial F(P)}{\partial A_c}^T & \frac{\partial F(P)}{\partial A_p} & \frac{\partial F(P)}{\partial A_p}^T \end{bmatrix} \quad (16)$$

면  $J_F(P_k)^T J_F(P_k)$  또는  $J_F(P_k)^T J_F(P_k) + \lambda_k \text{diag}\{J_F(P_k)^T J_F(P_k)\}$ 의 역행렬을 구해야 한다. 입체 영상을 제작하는 SFM이나 SLAM과 같은 작업에서는 매우 많은 양의 3D point가 발생하는데, 따라서  $J_F(P_k)^T J_F(P_k)$ 의 크기가 매우 커지게 되고, 역행렬을 구하기 위한 계산량도 매우 많아진다. 그래서 역행렬 계산량을 줄이기 위해 Schur complement를 이용할 수 있다. 각  $i$ 번째 camera parameter들을 모아놓은 열벡터를  $A_{c_i}$ 라고 하고, 마찬가지로 각  $i$ 번째 3D point의 3차원 좌표를  $A_{p_i}$ 라고 하자. 모든  $A_{c_i}$ 를 모아놓은 열벡터를  $A_c$ , 모든  $A_{p_i}$ 를 모아놓은 열벡터를  $A_p$ 라고 한다. 이때, 식 (15)처럼  $J_F(P)$ 를 각각 camera parameter에 대한 부분과 3D point에 대한 부분으로 나눌 수 있다.

$$J_F(P) = \begin{bmatrix} \frac{\partial F(P)}{\partial A_c} & \frac{\partial F(P)}{\partial A_p} \end{bmatrix} \quad (15)$$

따라서,  $J_F(P)^T J_F(P)$ 를 식 (16)처럼 4개의 블록행렬로 나눠서 Schur complement를 적용할 수 있다.

#### 4. Bundle Adjustment 적용 실험

<그림 5> ~ <그림 8>은 Bundle Adjustment 적용 여부만 달리하여 이미지 stitching을 수행한 결과들을 비교한 것이다. Bundle Adjustment 최적화 방법으로는 Levenberg-Marquardt 방법[3]을 사용하였다. <그림 5>를 보면, Bundle Adjustment를 적용하지 않았을 때에는 계단이 크게 어긋나 있지만, Bundle Adjustment를 적용했을 때는 자연스럽게 stitching 되었다. 마찬가지로 <그림 6>과 <그림 7>에서 Bundle Adjustment를 적용하지 않은 결과 영상은 크게 어긋난 부분이 있지만, Bundle Adjustment를 적용하면 이러한 부분이 개선되는 것을 확인할 수 있다. 특히, <그림 8>은 Bundle Adjustment를



<그림 5> Bundle Adjustment 적용 여부에 따른 비교 실험 1



<그림 6> Bundle Adjustment 적용 여부에 따른 비교 실험 2



<그림 7> Bundle Adjustment 적용 여부에 따른 비교 실험 3



<그림 8> Bundle Adjustment 적용 여부에 따른 비교 실험 4

적용하지 않은 결과 영상이 매우 부자연스러운데, 이 역시 Bundle Adjustment를 통해 크게 개선되었다. 이처럼, Bundle Adjustment 적용 여부에 따라서 결과 영상의 품질이 크게 달라지는 것을 확인할 수 있다.

## IV. 결론

본 글에서는 Metaverse에 활용될 수 있는 영상 제작에 서의 Bundle Adjustment에 대하여 설명하였다. Bundle Adjustment를 통해 camera parameter 및 3D point 좌표를 최적화함으로써 error를 최소화할 수 있으며, 따라서

결과 영상의 품질을 크게 개선할 수 있다. 최적화 알고리즘으로는 Gradient Descent, 가우스-뉴턴법, Levenberg-Marquardt 등의 방법이 있고, 주로 Gradient Descent와 가우스-뉴턴법의 장점을 모두 가지고 있는 Levenberg-Marquardt 방법이 많이 사용된다. 이처럼 정밀한 결과 영상을 얻기 위해서는 Bundle Adjustment를 수행해야 하지만, Bundle Adjustment는 계산량이 많은 작업이므로 결과 영상 생성 시간이 증가하게 된다. 특히, 매우 많은 수의 3D point를 가지는 SFM이나 SLAM과 같은 작업에서는 이러한 문제점이 더 두드러진다. 따라서, Bundle Adjustment의 속도를 증가시키기 위한 방법들이 연구되고 있다.

## 참 고 문 헌

- [1] SCHONBERGER, Johannes L.; FRAHM, Jan-Michael. Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. p. 4104-4113.
- [2] GAO, Xiang; ZHANG, Tao. Introduction to visual SLAM: from theory to practice, Springer Nature, 2021.
- [3] MARQUARDT, Donald W. An algorithm for least-squares estimation of nonlinear parameters. Journal of the society for Industrial and Applied Mathematics, 1963, 11,2: 431-441.
- [4] LEVENBERG, Kenneth. A method for the solution of certain non-linear problems in least squares. Quarterly of applied mathematics, 1944, 2,2: 164-168.
- [5] GAVIN, Henri P. The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems. Department of civil and environmental engineering, Duke University, 2019, 19.
- [6] M. Powell. A Hybrid Method for Nonlinear Equations. In P. Rabinowitz, editor, Numerical Methods for Nonlinear Algebraic Equations, pages 87-144. Gordon and Breach Science, London, 1970.
- [7] USENKO, Vladyslav, et al. Visual-inertial mapping with non-linear factor recovery. IEEE Robotics and Automation Letters, 2019, 5,2: 422-429.
- [8] WEBER, Simon, et al. Power Bundle Adjustment for Large-Scale 3D Reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. p. 281-289.
- [9] INDELMAN, Vadim; ROBERTS, Richard; DELLAERT, Frank. Incremental light bundle adjustment for structure from motion and robotics. Robotics and Autonomous Systems, 2015, 70: 63-82.
- [10] YE, Zhichao, et al. CoLi-BA: Compact Linearization based Solver for Bundle Adjustment. IEEE Transactions on Visualization and Computer Graphics, 2022, 28,11: 3727-3736.
- [11] BROWN, Matthew; LOWE, David G. Automatic panoramic image stitching using invariant features. International journal of computer vision, 2007, 74: 59-73.

## 저 자 소 개



### 최 건 찬

- 2018년 ~ 현재 : 세종대학교 전자정보통신공학과 학사과정
- ORCID : <https://orcid.org/0009-0004-9156-8853>
- 주관심분야 : 영상 신호처리, VR



### 한 종 기

- 1992년 : KAIST 전기및전자공학과 공학사
- 1994년 : KAIST 전기및전자공학과 공학석사
- 1999년 : KAIST 전기및전자공학과 공학박사
- 1999년 3월 ~ 2001년 8월 : 삼성전자 DM연구소 책임연구원
- 2001년 9월 ~ 현재 : 세종대학교 전자정보통신공학과 교수
- 2008년 9월 ~ 2009년 8월 : University California San Diego (UCSD) Visiting Scholar
- ORCID : <https://orcid.org/0000-0002-5036-7199>
- 주관심분야 : VR, 3D 영상신호 구현, 비디오 코덱, 영상 신호처리, 정보 압축, 방송 시스템