

이 문서는 공대-치대 학제 간 융합 연구 지원사업의 일환으로 진행된 파노라마방사선장비의 평가를 위한 볼 팬텀 영상 분석 소프트웨어의 기술 문서입니다. 제작된 소프트웨어의 이론적 배경과 각 개별 기능의 원리를 밝힘으로써 이후 유지 보수 및 추가 개발의 편의성을 높이고, 사용자들의 이해를 도모하기 위한 목적으로 작성되었습니다.

## 1. OpenCV (Open Source Computer Vision Library ) <sup>1)</sup>

OpenCV는 오픈소스 컴퓨터비전 라이브러리입니다. 영상인식과 처리에 관해 연구하는 컴퓨터비전 분야에서 OpenCV는 영상관련 라이브러리로서 사실상 표준의 지위를 가지고 있습니다. 윈도우, 리눅스, 맥OS, 안드로이드 등의 다양한 플랫폼을 지원하고, 프로그래밍 언어 중 C++, Python을 공식적으로 채택하여 지원합니다. 그밖에 C#, Java, Objective-C와 같은 다양한 프로그래밍 언어를 통해서도 사용이 가능하고, MATLAB 등의 프로그램과 연계가 가능합니다. 2500개 이상의 최적화된 영상처리 알고리즘과 OpenCV의 BSD 라이선스<sup>2)</sup>를 적용 덕분에 연구적 목적 이외의 상업적 이용에도 가장 보편화된 영상처리 라이브러리로 볼 수 있습니다.

## 2. C#

기존에 작성되었던 파노라마방사선장비 성능평가 프로그램은 MATLAB으로 작성된 까닭에 MATLAB 사용이 가능한 사용자에게만 제한적으로 사용이 가능하였습니다. 이번 연구 지원사업의 목적 중 하나로써 성능 평가프로그램의 임상 현장에서의 편리한 사용이 있었기 때문에, 본 프로젝트에서는 윈도우 환경에서 동작하는 윈도우 프로그램을 작성을 위해 프로그래밍 언어 중 C#을 이용하여 진행되었습니다. C#은 마이크로소프트에서 개발한 윈도우 프로그램 개발 및 실행환경인 닷넷 프레임워크(.NET Framework)에서 동작되므로, 이번에 제작된 소프트웨어는 닷넷 프레임워크가 설치된 윈도우 환경이라면 이상 없이 실행될 수 있습니다.

MATLAB 프로그램에서는 영상분석을 위해 추가적인 툴 박스(Image Processing Toolbox)가 설치되어야 합니다. C#에서는 OpenCV를 지원하는 다양한 라이브러리가 존재하지만, 그 중에서 가장 활발하게 사용되는 OpenCVSharp을 통해서 프로그램을 작성하였고, OpenCV의 다양한 알고리즘을 활용하여 파노라마방사선장비성능 평가를 위해 촬영된 볼 팬텀 사진을 분석할 수 있었습니다.

## 3. 사용된 외부 패키지

소프트웨어 작성 중에 분석결과의 정확성을 높이고 사용자의 편의를 도모하기 위해서 외부 라이브러리를 활용하였습니다. 본 소프트웨어의 범용적인 사용을 위해 활용된 라이브러리들의 라이선스를 밝히고, 해당 라이브러리가 어떤 역할을 담당하는 지를 짧게 설명하고자 합니다.

### 1) OpenCVSharp3

- 라이선스 : BSD-3-Clause
- 관련 페이지 : <https://github.com/shimat/opencvsharp>

OpenCVSharp 라이브러리는 닷넷프레임워크 환경에서 OpenCV를 활용할 수 있도록 하는 라이브러리입니다.

---

1) <https://opencv.org/>

2) <https://opensource.org/licenses/BSD-3-Clause>

## 2) Math.Net.Numerics

- 라이선스 : MIT/X11 License
- 관련 페이지 : <https://numerics.mathdotnet.com/>

Math.Net Numerics는 과학 기술 분야에 자주 쓰이는 수학적 계산에 대한 알고리즘을 닷넷 환경에서 사용할 수 있게 해줍니다. 본 소프트웨어에서는 영상의 해상도 계산을 위한 기능인 MTF 함수 계산에 필수적으로 시행되는 푸리에 변환(Fourier Transform)에 해당 라이브러리를 사용하였습니다.

## 3) fo-dicom.Desktop

- 라이선스 : Microsoft Public License (MS-PL)
- 관련 페이지 : <https://github.com/fo-dicom/fo-dicom>

Dicom은 의료영상의 표준 포맷입니다. Image Processing Toolbox를 사용하는 MATLAB의 경우 특별한 처리 없이 Dicom 포맷의 영상을 읽을 수 있지만, 다른 프로그래밍 언어로 작성된 소프트웨어의 경우에는 특별한 라이브러리의 도움을 받아서야 해당 영상을 읽고 분석할 수 있습니다. 이번 프로젝트에서는 닷넷 환경에서 Dicom영상을 처리할 수 있는 C# 라이브러리 중에서 가장 가볍고 사용이 많이 되고 있는 fo-dicom-Desktop을 활용하였습니다.

## 4. 소프트웨어의 각 기능에 대한 기술적 설명

이하의 내용들은 이번 프로젝트의 해당 기능들을 개발하기 위해 활용한 방식과 새롭게 개발한 알고리즘들에 관한 개괄적인 설명을 담고 있습니다. 좀 더 구체적인 내용과 코드의 구성을 알고 싶으신 분들은

- ① 개발을 담당한 서울대 공대 컴퓨터공대 컴퓨터공학부 분산시스템 연구실 소속의 경우, 파일공유 서버의 해당 프로젝트 디렉토리를 참고해 주시고,
- ② 일반 사용자의 경우 프로젝트 공개 페이지<sup>3)</sup>를 참조하시거나 개발자에게 메일<sup>4)</sup>을 보내주시면 답변 드리도록 하겠습니다.

### 1) 분석 영상의 선택

볼 팬텀을 촬영한 파노라마방사선영상에서 전체 사진을 분석 대상으로 삼을 경우, 분석 영역이 아닌 영상부분으로 인해 악골 도면과의 매칭에 오류가 생길 수 있습니다. 좀 더 구체적으로 살펴보면, 초점이 맞는 분석 영역이 아닌 부분은 볼 팬텀의 볼로 인식되는 정도가 규칙적이지 않습니다. 따라서 전체 영상을 사용할 경우, 사용자가 도면과의 매칭 기준점을 지정해주지 않으면 매칭의 정확성을 담보할 수 없어서 분석결과를 신뢰할 수 없게 됩니다.

따라서 이번 프로젝트에서는 매칭 기준점을 지정하는 것보다 초점이 맞는 분석영역을 선택하는 것이 좀 더 사용자에게 편하고 높은 정밀도를 요구하지 않는다는 이유를 들어, 분석 전에 사진의 분석 영역을 선택할 수 있도록 하였습니다. 사용자가 마우스 입력을 통해 선택한 영역에 대해서 분석을 진행하는 방식을 취하고, 전체 사진과 다른 영역에 해당되는 분석사진을 저장하여서 이후에 추가적인 확인이 가능하도록 하였습니다. 더불어 이 과정에서 Dicom 포맷의 파일이 분석대상일 경우, 이후에 활용할 수 있는 Dicom의 메타데이터는 저장하고, 분석 영상은 전체적인 분석과정의 통일성을 위해서 파일 포맷을 변경하였습니다.

---

3) <https://github.com/hayunjong83/phantom>

4) hayunjong83@gmail.com

## 2) 볼 인식과 악골 도면과의 매칭 후 매칭 정보 확인하기

해당 소프트웨어의 핵심은 촬영영상의 볼 팬텀 볼 위치를 악골 도면의 정확한 위치에 매칭시키는 것으로 생각할 수 있습니다. 우선 첫 번째로 촬영영상의 볼 인식에서 별다른 처리 없이 기존의 OpenCV 원 인식 알고리즘을 선택하는 경우, 촬영영상의 볼 인식률이 상당히 낮게 나옵니다. 전체적인 볼 인식이 낮은 경우, 매칭 될 대상이 적다는 문제에 그치지 않고 정확하지 않은 매칭 기준이 설정된 탓에 도면과의 매칭의 신뢰성이 낮고 추가적인 분석 기능들이 잘 못된 결과를 가지게 됩니다.

이러한 까닭에 이번 프로젝트에서는 그 이유를 분석하고 두 가지 추가적인 방식으로 이 문제를 해결하였습니다. 볼 인식이 저조한 것은 촬영 영상의 전체적인 밝기가 고르지 않기 때문입니다. 기존의 OpenCV에서는 가장 높은 픽셀값과 가장 낮은 픽셀값 사이에 평균값과 같은 방식으로 기준을 정한 후, 기준보다 높은 지점들은 분석의 대상으로 보고 기준보다 낮은 지점들은 배경에 해당하는 영역으로 보아 무시합니다. 따라서 전체적인 밝기가 고르지 않고, 어두운 영역이 넓게 존재하는 영상의 경우에는 볼 인식이 낮은 문제가 발생하였습니다. 이를 해결하기 위해서 분석 영상의 전체에서 기준 픽셀값을 정하는 방식 대신에 영상을 10개의 균등한 영역으로 나누고 각 영역마다 기준값을 구하여 분석 대상을 확장하는 방식을 취하였습니다. 이러한 방식을 통해서 볼의 인식률을 크게 높일 수 있었습니다.

두 번째로 볼이 인식되지 않는 것은 촬영영상에서의 볼이 OpenCV가 원으로 인식하는 전통적인 의미의 원이나 타원이 아닌 모양으로 촬영되는 경우가 흔히 발생하였기 때문입니다. 사각형에 가까운 모양의 경우 볼이 아닌 것으로 판단되거나 제거해야하는 노이즈로 분리되는 문제로 인해 볼 팬텀의 경계영역은 인식이 잘 되지 않는 경우가 일반적이었습니다. 두 번째 문제를 해결하기 위해서 앞선 과정에서 볼로 인식된 영역으로부터 분석 선을 연장하여 굵고, 분석선 상에서 만나게 되는 높은 픽셀값의 영역을 만나게 되었을 경우, 그 모양을 분석하는 대신 인접한 픽셀영역의 수를 확인하도록 하였습니다. 인접한 픽셀 영역의 수가 특정값을 넘는 경우는 뭉쳐있는 넓이를 갖는 대상이 촬영된 경우로 생각할 수 있다고 판단하였기 때문입니다. 인접한 픽셀영역을 확인하는 방식은 너비우선탐색(bfs: breadth-first-search) 알고리즘을 활용하였습니다. 결과적으로 두 가지 추가적인 분석방식을 통해 촬영영상의 전체적인 인식률을 크게 높일 수 있었습니다.

촬영 영상에서 볼을 인식한 후에 악골 도면과 매칭하기 위해서는 인식된 볼들 행(row)과 열(column)로 분류하는 작업이 필요합니다. 좀 더 상세한 알고리즘에 대해서는 앞서 언급된 방식을 통해 문의를 부탁드립니다. 여기서는 매칭의 순서와 주요 개념에 대한 내용을 설명합니다. 우선 인식된 볼들의 Y좌표와 X좌표를 동시에 고려하여 정렬합니다. 이 때 Y좌표의 값을 중심으로 그룹을 만들어 하나의 행(row)으로 분류되는 볼들을 선택합니다. 이 과정에서 그룹에서 벗어난 볼들에 대해 추가적인 처리과정을 통해 전체 볼들을 정렬된 행(row)의 집합으로 분류합니다. 각 행의 집합에서 다시 X좌표 값을 중심으로 정렬하여, 열(column)로 분류하고 최종적으로 행렬에 매칭 시키는 과정을 준비합니다. 정렬된 행(row)을 하나씩 탐색하면서 X좌표가 가장 큰 볼들을 가져오는 방식으로 하나의 볼에 대한 열(column)값을 지정해주고 같은 열에 해당하는 볼을 분류할 수 있습니다. 이런 수평분류와 수직분류의 두 과정을 통해 영상의 각 볼들은 행과 열의 인덱스를 가지는 객체로 변환될 수 있고, 그 인덱스에 맞춰 악골 도면의 볼 위치에 매칭 시키는 방식으로 영상과 도면의 매칭을 완료합니다.

촬영 영상의 촬영방식 및 상태에 따라 이러한 처리과정에도 불구하고 매칭이 제대로 되지 않는 경우가 발생할 수 있습니다. 따라서 이러한 과정의 수정을 위해 현재 영상에 대한 매칭 정

볼을 구체적인 분석 결과 이전에 보여줄 수 있는 기능을 소프트웨어에 추가하였고, 사용자는 분석의 오류나 정확성이 존재할 수 있는 영역을 인지할 수 있게 됩니다.

### 3) 각 분석 기능의 기술적 특이점

#### - Image Layer

볼 팬텀의 상층영역을 판단할 수 있도록 분석한 Image Layer기능은 제작된 소프트웨어의 핵심 기능으로 볼 수 있습니다. 현재 촬영영상에서 초점이 맞는 영역을 의미하는 용어로서 포컬 레이어(focal layer)를 혼용하여 표현하였습니다. 기존의 MATLAB 프로그램에서는 인식된 볼의 너비와 높이의 비율을 측정한 후, 가장 원에 가까운 볼들을 초점영역으로 판단하는 방식을 택하였습니다. 하지만 이 방식의 경우 해당 소프트웨어에서 볼에 해당하는 영역을 판단하는 방법에 크게 의존되어 있었고, 볼의 일부분만 원으로 인식되었을 경우 흐린 볼 영상이 초점 영역으로 판단되거나, 너무 많은 수의 볼들이 초점이 맞는 영역에 속한다고 판단되는 문제가 발생하였습니다.

따라서 기존의 분석 프로그램과의 일관성을 위해서 너비/높이 비율에 따른 초점영역을 남겨 두면서, 볼 경계에서의 픽셀값들의 변화율을 살펴보는 새로운 방식으로 포컬 레이어 영역을 결정하였습니다. 경계의 픽셀값이 급격히 변화한다는 것은 선명한 경계를 가진다는 의미이므로 변화율을 정렬한 후 상위 볼들을 초점 영역으로 분류하였습니다.

#### - MTF

전체 영상의 해상도를 미뤄 짐작할 수 있는 MTF 수치는 앞서 포컬 레이어를 구하는 이유와 동일한 까닭에서 볼 팬텀 촬영영상의 영역에 따라서 상이한 값을 가집니다. 따라서 볼 팬텀의 주요 영역에 해당하는 4개의 볼과 사용자가 직접 분석하기를 희망하는 선택된 볼에 대해 MTF 평가를 진행합니다. MTF(Modulation Transfer Function)으로 분석 대상의 픽셀값의 거리에 따른 변화를 나타내는 ESF(Edge Spread Function)과 ESF함수의 변화율을 나타내는 LSF(Line Spread Function)함수, 그리고 최종적으로 이 LSF함수를 푸리에 변환하는 방식의 일반적인 알고리즘을 채택하여 이뤄졌습니다. 참고할 수 있는 C#으로 구현된 기존 MTF 모듈이 존재하지 않은 까닭에, 프로젝트 전에 인계받았던 MATLAB프로그램의 MTF 알고리즘과 동일한 방식으로 본 기능을 구현하였습니다.

#### - Central Ray

central ray는 방사선촬영영상의 촬영방향을 미뤄 짐작할 수 있는 기능입니다. 촬영 영상의 가장 상위의 볼로부터 수직선을 내려 그었을 때, 가장 아래쪽 볼들 사이에서 인접한 볼들을 결정하고, 그 볼들 사이의 인접비율을 구합니다. 이렇게 출발점과 인접한 볼들 및 그 비율을 악골 도면으로 옮김으로써 해당지점에서의 촬영방향 및 전체적인 촬영방향의 비율을 확인할 수 있습니다. 짧은 central ray는 촬영영상의 정보를 그대로 도면으로 옮긴 것이고, 긴 central ray는 간단한 일차 직선의 계산을 통해서 특정 길이의 직선들로 확장한 것입니다. 이때 직선의 길이는 인접한 central ray가 중첩될 수 있는 길이로 임의로 지정하고, 이 때 인접한 선과의 교점은 촬영기기의 rotational center로 유추할 수 있는 정보로 볼 수 있으므로 추가적으로 central ray정보와 함께 표기할 수 있도록 하였습니다.

#### - Image gain velocity

영상획득속도의 기능은 가장 가운데 행(row)의 볼들 간격으로부터 영상이 촬영되는 속도의 변화를 알아볼 수 있는 기능입니다. 아직 그 활용을 위한 방법이 결정되지 않았기 때문에, 단순히 볼들 간격을 그래프로 표현하여 나타내었습니다.