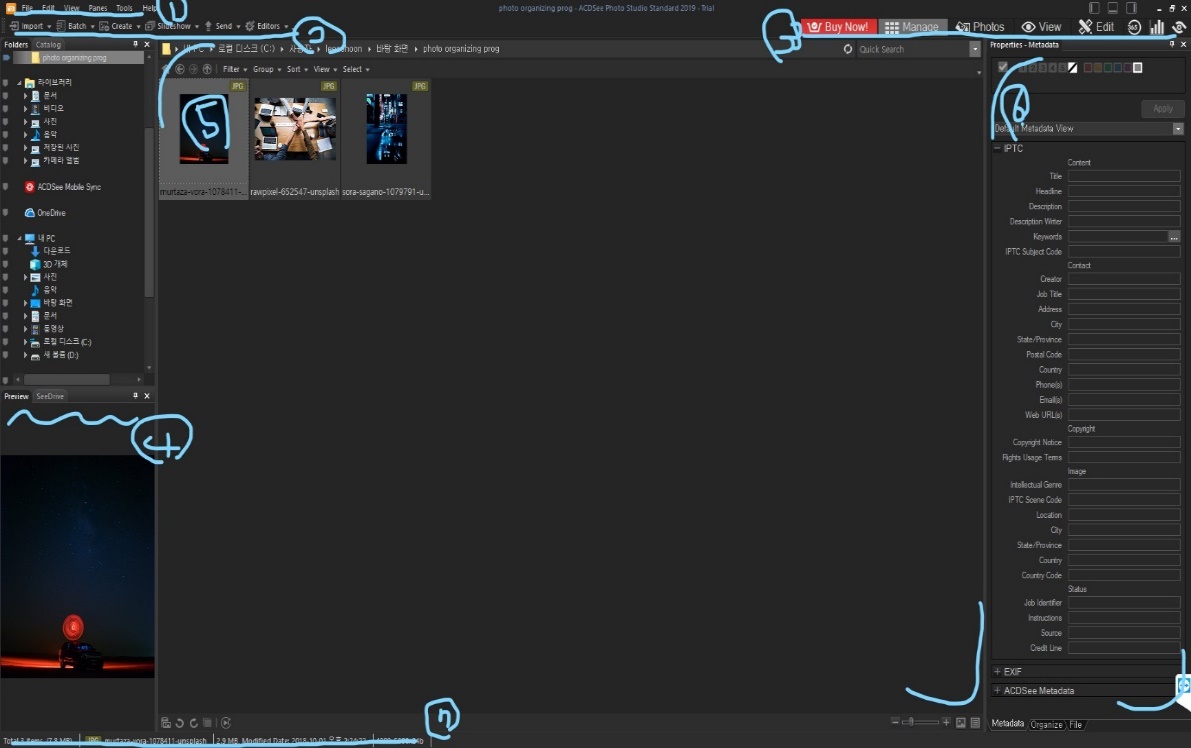
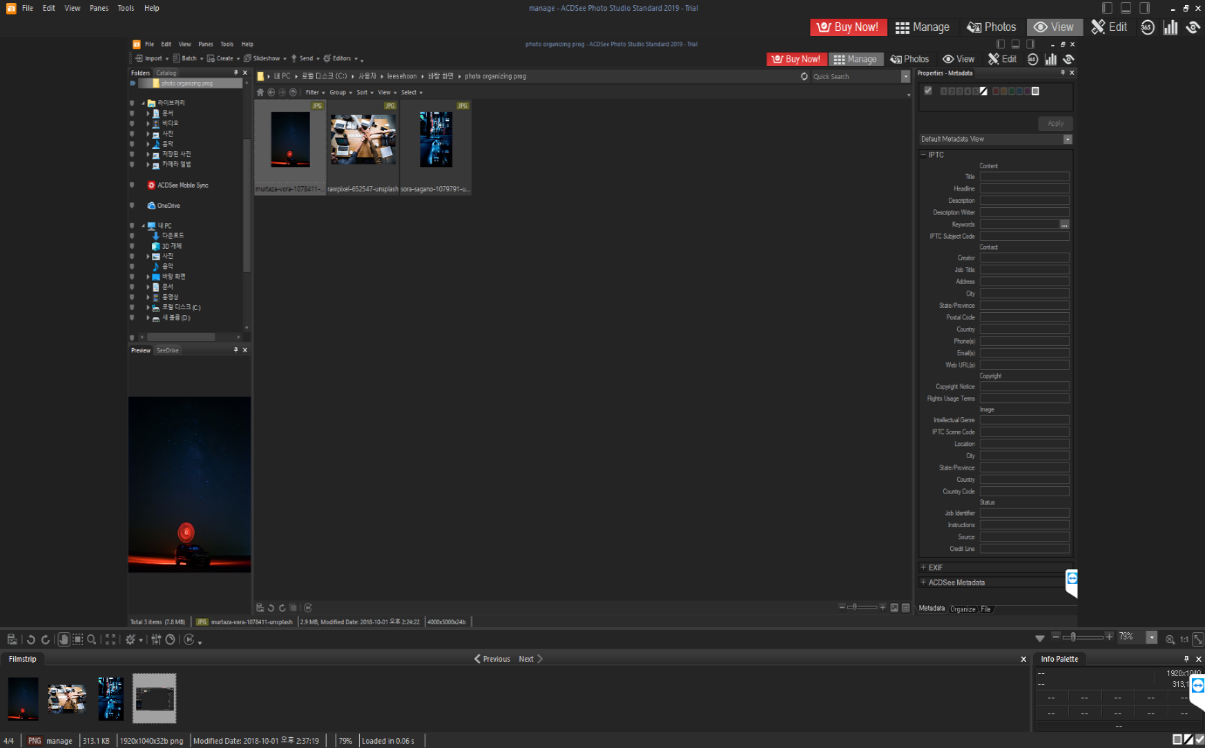
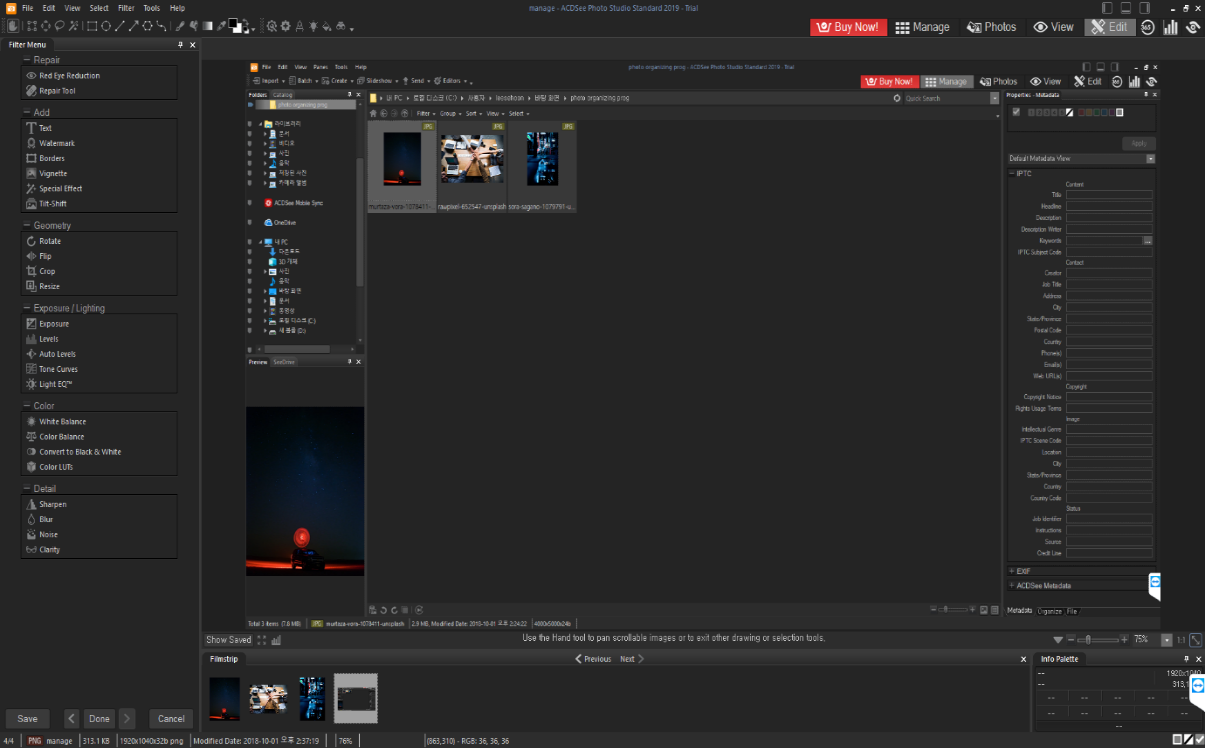
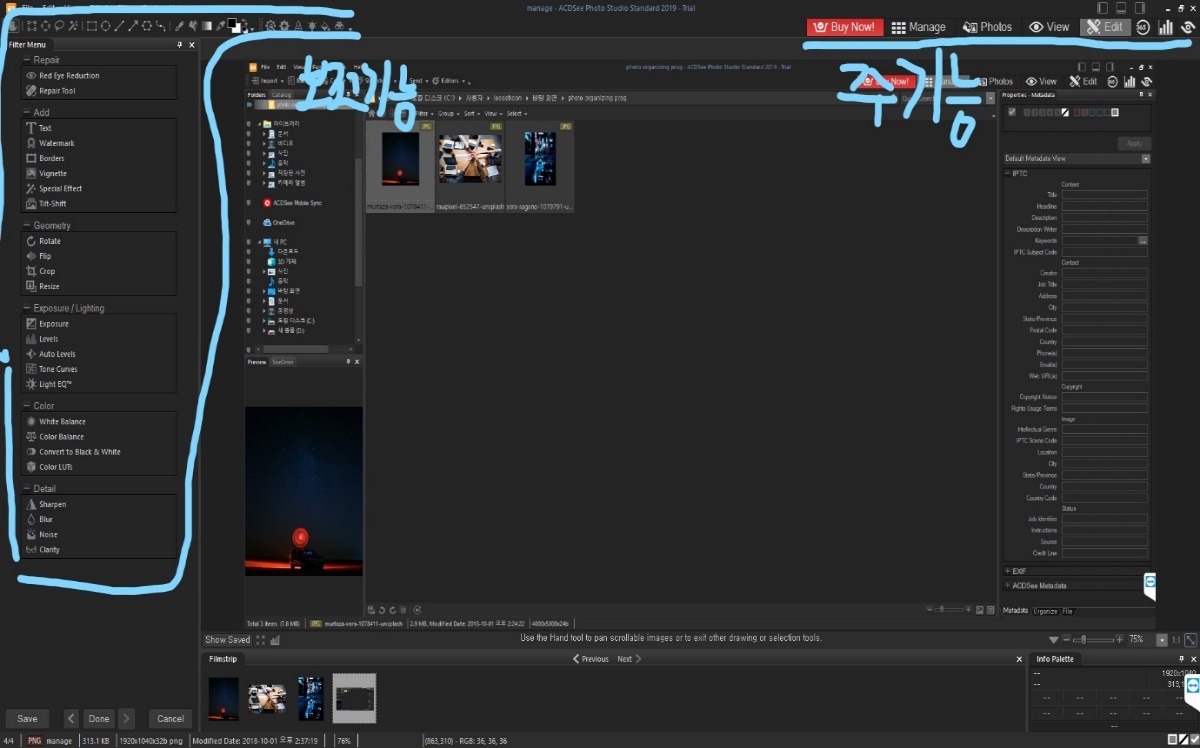
1. ACDSee는 사진 보정 소프트웨어로 수평적, 수직적 인터페이스를 잘 구성했다. 수평적 인터페이스는 한 화면에 있는 상호 작용의 수이고 수직적 인터페이스는 특정 업무를 하기 위해 거치는 상호작용의 수라고 정의한다. 1번 질문에서는 수평적 인터페이스를 원칙, 가이드라인, 이론을 바탕으로 평가해보고자 한다.

ACDSee의 수평적 인터페이스는 1)최상단, 2)차상단-좌측, 3)차상단-우측, 4)메인-좌측, 5)메인-중앙, 6)메인-우측, 7)메인-좌측, 하단으로 구성되어 있다. 최상단에는 시스템에 필요한 상호작용, 차상단-좌측은 기능에 필요한 단축버튼, 차상단-우측은 주요 기능, 메인-좌측은 구체적 기능, 메인-중앙은 주요 화면, 메인-우측은 사진 파일의 세부 사항, 최하단은 사진 파일의 간략한 설명을 보여준다.

먼저, 원칙적으로 좋은 HCI를 요약하자면 ‘사용자를 편하게 만들어라’ 라고 생각한다. 그런 측면에서 ACDSee의 인터페이스는 확실히 전문적인 사진 보정이 필요한 사람들을 초점으로 맞춰져 있다. 예를 들어 메인 기능에 view만을 위한 기능이 따로 있는 것, 사진의 세부 사항을 메인-우측에서 보여주는 것 등은 전문적인 사진 보정 작업을 하는 사람을 위한 것이다.

또한, 일관성을 유지해야하는 인터페이스를 잘 분리해 놓았다. 예를 들면 최상단, 차상단-우측, 최하단 같은 경우에는 사진 보정과 프로그램 사용에 지속적으로 필요한 내용이기 때문에 고정되어 있다. 다만, 뷰기능의 화면에서 일관성이 아쉬운 부분이 있다. 차상단 좌측에 있으면 좋았을 기능이 하단에 있는 것이다. 그러나 이것 또한 하단의 사진리스트와 가깝게 배치하기 위해 일관성보다는 편의성을 택했다고 생각한다.

구체적인 가이드라인도 잘 지키고 있다. 앞서 말했듯이 중요한 기능과 정보는 화면에서 고정시켰다. 또한 메인-좌, 우측의 정보들이 중요도와 사용률에 따라서 배치되어 있다. 다만, 특정 기능을 찾기 위해서 일일이 클릭해야 하는 수고로움이 있으므로 기능을 바로 찾아서 시행할 수 있는 검색창을 추가하는 것도 괜찮다고 생각한다.

이론적인 부분도 잘 반영하고 있는 인터페이스다. 노만의 행동 과정 이론은 유저의 의도와 행동이 일치하고, 시스템의 표현과 유저의 기대가 일치해야 된다는 결론을 내린다. 따라서, 차상단-우측에는 중요한 메인기능의 아이콘과 텍스트 모두 표현해 놓았다. 뿐만 아니라 메인기능과 보조기능을 각각 붙여 놓아서 시각적으로 청크화 시켜놓았다.

1. 사진 보정 프로그램을 통해 보정을 완료하기 위해서는 사용자는 몇 번 선택을 해야 한다. 하지만 결정의 대부분은 yes or no로 대답하기 곤란하거나 부적절한 경우이다. 예를 들어 어떤 보정을 할 것인가에 따라 보조기능을 선택할 때, 모든 기능에 대해 매번 가부를 묻는 것 보다 모든 기능 중에 선택하는 것이 낫다. 또한 사진의 RGB 값을 조정한다고 했을 때, 255\*255\*255의 경우의 수를 모두 묻는 것 보다 직접 입력하는 편이 낫다.

따라서 Decision Tree를 그리는 것이 부적절 하다. Decision Tree를 그리기 위해서는 Training Data와 이를 바탕으로 결정의 기준이 되는 Splitting Attributes가 필요하다. 하지만 사진 보정은 너무 많은 Training Data가 필요하고 그에 따른 Decision Tree의 높이도 길어진다.

하지만 몇 가지 방법을 추가해서 어마어마한 트리를 빠르게 처리하는 것을 생각해보았다. 먼저, Grouping을 통해 경향성을 형성해주는 것이 가능하다. 1~255 값을 모두 가부를 묻지 않고 1~81, 82~163, 163~255로 나누고 그 안에서 다시 나누어서 묻는다면 훨씬 빠를 것이다. 뿐만 아니라 여러 번 Decision이 끝난다면 선택된 횟수에 따라 많이 선택된 Splitting Tree를 먼저 결정할 수 있도록 트리의 위로 옮기는 것도 좋다고 생각한다.

1. 사진에 텍스트를 추가하는 과정을 Goms model에 적용해 보기로 하였다. 타자는 비숙련자, 판단을 위한 생각은 하지 않고, 한 손을 사용하는 것을 가정하였다. 예측시간은 10.65초였으며, 실제 시간은 13초 정도 걸렸다.

예측시간과 실제 시간 차이에는 역시 M’s time의 차이가 있었다. 실제 시간 관측을 위해 여러 번 같은 과정을 반복하여 숙달 후에 촬영을 했다. 그럼에도 불구하고 아이콘을 찾고 클릭할 때 마다 생각하는 시간이 필요했다. 오히려, 마우스에서 키보드로 옮기는 시간은 덜 걸렸던 것 같다.

이 실험을 하면서 은행 사이트의 마우스를 통해 무작위 배열된 키보드 입력의 비효율성과 핸드폰 생체 보안 인식의 편리함에 대해 비교해보는 계기가 되었다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 순서 | 행동 | KLM | 예측시간 | 실제시간 |
| 1 | 바탕 화면 폴더 가리키기 | P | 1.1 |  |
| 2 | 바탕 화면 폴더 클릭하기 | B | 0.1 |  |
| 3 | 사진 폴더 가리키기 | P | 1.1 |  |
| 4 | 사진 폴더 클릭하기 | B | 0.1 |  |
| 5 | 사진을 가리키기 | P | 1.1 |  |
| 6 | 사진을 클릭하기 | B | 0.1 |  |
| 7 | Edit 기능을 가리키기 | P | 1.1 |  |
| 8 | Edit 기능을 클릭하기 | B | 0.1 |  |
| 9 | Text 삽입 기능을 가리키기 | P | 1.1 |  |
| 10 | Text 삽입 기능을 클릭하기 | B | 0.1 |  |
| 11 | Text 칸을 가리키기 | P | 1.1 |  |
| 12 | Text 칸을 클릭하기 | B | 0.1 |  |
| 13 | “hello”를 입력하기 | H+KKKKK | 1.85 |  |
| 14 | Done을 가리키기 | H+P | 1.5 |  |
| 15 | Done을 클릭하기 | B | 0.1 |  |
|  | | 합계(단위:초) | 10.65 | 13 |

참고 영상: https://youtu.be/cE1D5UyuAhE

1. 갤럭시 노트 시리즈의 다른 핸드폰과 다른 큰 특징 중 하나는 스타일러스 펜이다. 이 펜은 EMR, Electro-Magnatic Resonanace라는 기술을 사용하였다. EMR 기술은 디바이스 컨트롤러가 자기장을 만들어 펜으로 주파수를 보낸다. 이를 감지한 펜은 무선 주파수를 다시 전달한다. 1)이 때 디바이스 컨트롤러는 전송을 중지하고 청취모드로 돌아서며 여러 지점의 신호 세기를 측정해 펜의 위치, 시기와 세기를 계산하여 레지스터에 저장하고 2)interrupt를 일으킨다. 3)그러면 실행 중인 user process는 interrupt handler를 실행한다. 4)핸들러는 디바이스 독립적 소프트웨어의 스케쥴링를 통해 해당 디바이스 드라이버를 호출한다. 5)디바이스 드라이버는 6)해당 디바이스 컨트롤러의 정보를 읽어서 7)디바이스 독립적 소프트웨어에 보내게 된다.

