2021학년도 수능 3D 합성 영상 지문 분석서

기출짓는늙은이(오르비)

최대한 실전에서 할 수 있는 사고 위주로 적었으며 실제 분석이 이것보다 더 자세히 들어가는 것을 염두하며 읽으시길 바랍니다.

2021수능 [34~37](독서)

> 최근의 3D 애니메이션은 섬세한 입체 영상을 구현하여 실물을 촬영한 것 같은 느낌을 준다. 실물을 촬영하여 얻은 자연 영상을 그대로 화면에 표시할 때와 달리 3D 합성 영상을 생성, 출력하기 위해서는 모델링과 렌더링을 거쳐야 한다.

지문의 첫 문단에서 우리가 잡아야 할 것은 글의 방향성이다. 즉 맹목적으로 정보를 처리하려는 태도를 지양하고 지문이 우리에게 던지고자 하는 것이 무엇인지를 잡아야 한다. 당연히 글의 기본적 방향은 3D 합성 영상이 생성, 출력되는 과정에 대한 것, 여기에 더해 그 과정의 기술에 대한 것이다.

과거의 기출같은 경우 'A,B를 위해 C,D가 필요하다' 등의 문장에서 A와 C가 연결되고 B와 D가 연결되는 것을 어느 정도 도출할 수 있었다만 이제 그런 얕은 예측은 지양하자.

> ***모델링***은 3차원 가상 공간에서 물체의 **모양과 크기**, **공간적인 위치**, **표면 특성** 등과 관련된 고유의 값을 설정하거나 수정하는 단계이다. **모양과 크기**를 설정할 때 주로 3개의 정점으로 형성되는 삼각형을 활용한다. 작은 삼각형의 조합으로 이루어진 그물과 같은 형태로 물체 표면을 표현하는 방식이다. 이 방법으로 복잡한 굴곡이 있는 표면도 정밀하게 표현할 수 있다. 이때 삼각형의 꼭짓점들은 물체의 모양과 크기를 결정하는 정점이 되는데, 이 정점들의 개수는 물체가 변형되어도 변하지 않으며, 정점들의 상대적 위치는 물체 고유의 모양이 변하지 않는 한 달라지지 않는다. 물체가 커지거나 작아지는 경우에는 정점 사이의 간격이 넓어지거나 좁아지고, 물체가 회전하거나 **이동하는 경우**에는 정점들이 간격을 유지하면서 회전축을 중심으로 회전하거나 동일 방향으로 동일 거리만큼 이동한다. 물체 표면을 구성하는 각 삼각형 면에는 **고유의 색과 질감 등을 나타내는 표면 특성**이 하나씩 지정된다.

모델링이 '3차원 가상 공간에서' 이뤄지는 단계라는 것을 보고 우리는 낙듭을 즉시 해야만 한다. 애초에 모델링은 3D를 구현하는 기술이다. 때문에 당연히 3차원에서 이뤄지는 것이며 이를 하나의 정보로 분류하여 머리에 집어넣고자 하는 것은 독해에 문제가 있다는 것.

첫 문장 '모델링은 3차원 가상 공간에서~단계이다.'는 우리에게 모델링에 대한 가장 기본적인 설명을 한다. 그러나 우리는 여전히 모델링에 대해 모른다. 때문에 친절한 평가원께서 구체화된 문장을 준다. **글을 바르게 읽기 위해선 다음의 문장들이 첫 문장을 구체화함을 이용하여 모델링에 대한 이해를 해나가야 한다.**

(추가로, 설정하거나/수정하는 단계이기에 수정의 가능성이 있단 것도 가볍게 이해하자.)

3개의 정점으로 형성된 삼각형을 통해 물체의 모양,크기를 설정한단 정보는 직관적으로 이해

가 가능하다. 이 삼각형의 조합으로 그물 형태를 만들고 물체 표면을 표현한단 것. 이 문장에서 독자의 독해력이 판가름나는 순간이 일어난다. 배경지식이 없어도 저 정보를 온전히 이해해야만 하는데, 그 근거는 삼각형의 '조합'이라는 것이다. 즉 여러 개의 삼각형이라는 것. 매우 당연하게 수많은 작은 삼각형이 모여 물체를 구성한단 걸 떠올려야 한다.
그러니 표면의 복잡한 굴곡도 표현할 수 있는 것이고.
물체 표면을 구성하는 삼각형들의 각 꼭짓점들의 정의가 정점이며, 정점의 특성들을 잘 이해해야 한다. 이 역시 단순암기로 해결하려는 태도를 지양하자. 물체 고유의 모습이 변하지 않는 한 삼각형 자체의 모습도 변하지 않을 것이기에 당연히 정점(꼭짓점)의 상대적 위치가 변하지 않는다. 개수 역시 마찬가지. 물체가 단순히 커지고 작아지는 것 역시 삼각형의 모양은 그대로 유지하되 크기만 커지는 것이므로 정점간의 간격이 그에 따라 변화하며, 회전과 이동도 마찬가지로 삼각형의 회전과 평행 이동으로 이뤄진다. **수험생이 이 부분을 분석할 때 분명히 알아야 할 것은 '이때 삼각형의 꼭짓점들은~'부터 '~동일 거리만큼 이동한다.'까지의 문장이 아주 긴밀히 연결됐다는 점이다. 물체를 구성할 때 정점이 어떻게 작용하고 어떻게 변형되는지 친절히 알려주는 부분이다.**
각 삼각형 면엔 표면 특성이 지정된다. 역시 당연하다. 애초에 작은 삼각형의 조합으로 물체 '표면'을 표현한다고 했다. 당연히 삼각형 면엔 표면의 특성이 지정될 것.

한편 기출을 통해 기술 지문에서 자주 등장하는 패턴을 확인할 수 있다.(언제까지나 경험적으로 파악한 특성이기에 무조건적으로 따르는 것은 추천하지 않음.) 모델링이 1.물체의 모양과 크기 2.공간적 위치 3.표면 특성 에 관한 값을 설정, 수정하는 단계이기에 저 세 예시에 대한 구체화는 필연적으로 등장할 것이다. CD드라이브 지문에서 CD드라이브의 각 구성요소를 먼저 나열하고 그 후에 구성요소들의 작용을 구체화하는 것과 유사하다. (지문에서 특정 부분을 볼드체 처리한 이유이다.)

> **공간에서의 입체에 대한 정보인 이 데이터를 활용하여, 물체를 어디에서 바라보는가를 나타내는 관찰 시점을 기준으로 2차원의 화면을 생성하는 것이 렌더링이다.** 전체 화면을 잘게 나눈 점이 화소인데, 정해진 개수의 화소로 화면을 표시하고 각 화소별로 밝기나 색상 등을 나타내는 화솟값이 부여된다. 렌더링 단계에서는 화면 안에서 동일 물체라도 멀리 있는 경우는 작게, 가까이 있는 경우는 크게 보이는 원리를 활용하여 화솟값을 지정함으로써 물체의 원근감을 구현한다. 표면 특성을 나타내는 값을 바탕으로, 다른 물체에 가려짐이나 조명에 의해 물체 표면에 생기는 명암, 그림자 등을 고려하여 화솟값을 정해 줌으로써 물체의 입체감을 구현한다. 화면을 구성하는 모든 화소의 화솟값이 결정되면 하나의 프레임이 생성된다. 이를 화면출력장치를 통해 모니터에 표시하면 정지 영상이 완성된다.

모델링에 대한 설명을 마친 뒤엔 당연히 렌더링에 대한 설명이 등장한다.
"공간에서의 입체에 대한 정보인 '이 데이터'를 활용하여"/"물체를 어디에서 바라보는가를 나타내는 관찰 시점을 기준으로 '2차원의 화면'을 생성하는 것이 렌더링이다."
이번 단락의 첫 문장을 통해 우선 **렌더링은 모델링의 결과값(데이터)를 이용한단 점을 머리에 박아두자.** 또한 2차원의 화면을 생성한단 것을 당연하게 받아들이자. 지금 우리가 다루는 지문이 어떤 것인가? 3D 애니메이션이다. 3차원에서 구현한 물체를 화면을 통해 바라보기에 **당연히 2차원으로 물체를 표현해야 한다**. 그렇기에 2차원 화면을 쉽게 납득할 수 있는 것이다.
(3D 영상에 대한 지문이니 당연히 3차원일 것이라고 생각했는데...?라고 인식하고 들어갔다면 위의 생각

을 통해 모델링과 렌더링이 각각 다른 차원의 문제라는 것을 인지해야만 한다.)
+)입체적인 물체를 단면으로 표현해야하니 관찰시점이라는 기준이 개입할 수 밖에...같은 사람이어도 앞에서 보면 눈코입이 있으나 뒤에서 보면 아니다.

화면의 화소에는 밝기, 색상 등을 나타내는 화솟값이 부여된다. **여기서도 역시 모델링의 결과를 통해 렌더링이 일어남을 알 수 있다.** 밝기, 색상. 다른 말로? 표면 특성.
정말 친절한 평가원께서 '표면 특성을 나타내는 값을 바탕으로~화솟값을 정해 줌'이라 알려주셨다. 제발 놓치지 말자.
이렇게 화솟값을 결정하면 하나의 프레임이 생성되고, 화면출력장치를 통해 정지 영상을 만든다. 여기까지 읽었을 때 독자는 원하는 것은 다 알았다는 반응을 해야 한다. 다시 첫 문단으로 돌아가자. 우리가 보고자 하는 것을 잊지 말아야 한다. 우리는 지금 3D 합성 영상이 어떻게 생성되고 출력되는지를 알려고 했다. 모델링 렌더링 자체가 중요한 것이 아니라 그것들을 통해 과정을 보고자 하는 것이다. 그러므로 우선적으로 파악할 것은 전부 알아냈다.

> 모델링과 렌더링을 반복하여 생성된 프레임들을 순서대로 표시하면 동영상이 된다. 프레임을 생성할 때, 모델링과 관련된 계산을 완료한 후 그 결과를 이용하여 렌더링을 위한 계산을 한다. 이때 정점의 개수가 많을수록, 해상도가 높아 출력 화소의 수가 많을수록 연산 양이 많아져 연산 시간이 길어진다. 컴퓨터의 중앙처리장치(CPU)는 데이터 연산을 하나씩 순서대로 수행하기 때문에 과도한 양의 데이터가 집중되면 미처 연산되지 못한 데이터가 차례를 기다리는 병목 현상이 생겨 프레임이 완성되는 데 오랜 시간이 걸린다. CPU의 그래픽 처리 능력을 보완하기 위해 개발된 ㉠그래픽처리장치(GPU)는 연산을 비롯한 데이터 처리를 독립적으로 수행할 수 있는 장치인 코어를 수백에서 수천 개씩 탑재하고 있다. GPU의 각 코어는 그래픽 연산에 특화된 연산만을 할 수 있고 CPU의 코어에 비해서 저속으로 연산한다. **하지만 GPU는 동일한 연산을 여러 번 수행해야 하는 경우, 고속으로 출력 영상을 생성할 수 있다.** 왜냐하면 GPU는 한 번의 연산에 쓰이는 데이터들을 순차적으로 각 코어에 전송한 후, 전체 코어에 하나의 연산 명령어를 전달하면, 각 코어는 모든 데이터를 동시에 연산하여 연산 시간이 짧아지기 때문이다.

이 문단을 처리할 땐 기출분석의 누적에 따른 여유의 차이가 발생한다. 많은 기출 독서 지문에서 근본적인 원리, 정보 등이 서술된 후 추가적으로 관련 정보가 등장한다. 하지만 결국은 주제와 관련될 것이니...글의 방향인 '3D 합성 영상 생성 및 출력'에 맞추어 읽도록 하자.

우선 모델링/렌더링으로 생성된 프레임들을 순서대로 표시해 동영상을 만든다. 납득할 수 있는 정보다. 하나의 프레임을 표시하면 정지 영상일 것이고, 여러 프레임을 표시하면 정지가 아닌 움직이는 영상일 것이니까. 영상을 만들기 위한 프레임을 생성하기 위해 **모델링과 관련된 계산을 완료한 후 그 결과를 이용해 렌더링을 위한 계산을 한다!** 이 정도면 모델링의 결과값으로 렌더링을 한다는 것은 절대진리가 돼야 한다. 정말 친절한 평가원...우리가 여기서 추가적으로 집중해야할 것은 결국 3D 영상을 만드는 '과정'이기에 지금 이야기하는 것이 어떤 과정인지다. 현재 이야기하는 부분은 모델링과 렌더링이 끝난 후가 아닌, 모델링이 끝난 후 렌더링을 하기 위한 과정이다.

이때 정점 개수가 많거나 해상도가 높아 출력화소가 많을수록 연상 양이 늘어나 연산 시간이

길어지고, CPU는 그 과정에서 문제가 생긴다. 당연히 필연적으로 이를 보완하고자 등장한 GPU를 살피자. 그런데 여기서 맹목적으로 읽지 말고, 무엇을 보완하고자 하는지 생각하고 넘어간다. 현재 CPU가 가진 문제는 '데이터 연산을 하나씩 순서대로 수행하기 때문에/~프레임이 완성되는 데 오랜 시간이 걸린다.'는 것. 그렇다면 **그를 보완한 GPU는 당연히 오랜 시간에 대한 개선을 할텐데 그것이 어떻게 이뤄질까? 당연히 문제점의 원인을 개선하는 것 아닐까? 여기서 작은 힌트를 얻고 가는 것이다.**

GPU는 연산 및 데이터 처리 수행 장치인 코어를 아주 많이 가지며 그래픽 연산에 특화된 연산만 가능하고, CPU의 코어에 비해 저속이다. 여기서 의문이 들수도 있다. **문제점이 오랜 시간이었는데 저속으로 하면 어쩌라는 것인가? 그렇기에 대신 '동일한 연산을 여러 번 수행해야 하는 경우'에서 보완이 이뤄진다. 그 보완은 당연히 저속->고속인 것이고.**

우리가 이 부분에서 지문을 넘어 문제까지 바라보는 힘을 가져야만 한다. 분명히 CPU/GPU가 문제화될 것인데 그 초점은 GPU의 개선된 능력에 있다. 그 능력은 '동일한 연산을 여러 번 수행해야 하는 경우'로 한정된다. 그러니 선지를 보며 GPU가 적용되는 경우가 어떤 경우인지 확실히 체크하길 바란다.

다시 지문 해설로 넘어가자. 왜 저런 경우에선 고속인지에 대해 알면 된다.

한 번의 연산에 쓰이는 데이터들을 순차적으로 각 코어에 전송->전체 코어에 연산 명령어 하나를 전달->각 코어가 모든 데이터를 동시에 연산. 이 과정을 통해 연산 시간이 줄어드는 것을 받아들이자. 핵심은 한 번의 연산에 쓰이는 여러 데이터를 동시에 연산하는 것이다.

34. 윗글에 대한 이해로 적절하지 않은 것은?

① 자연 영상은 모델링과 렌더링 단계를 거치지 않고 생성된다.

:모델링과 렌더링을 설명하는 이유가 3D 영상을 마치 실물을 촬영한 듯 섬세하게 표현하기 위한 것이다. 자연 영상인 저 과정이 필요없다는 것. 설명하기도 입이 아플 정도다. 적절.

② 렌더링에서 사용되는 물체 고유의 표면 특성은 화솟값에 의해 결정된다.

: 틀리면 안된다. 틀렸다면 문장 독해 및 문장 연결에 문제가 있는 것이다. 지문에서 수없이 모델링을 바탕으로 렌더링이 이뤄짐을 보여주었다. 즉 모델링에서 지정한 표면 특성을 바탕으로 화솟값을 결정하는 것. 선지의 핵심 정보 순서가 완전히 뒤바뀌었다. 적절하지 않음. 정답.

③ 물체의 원근감과 입체감은 관찰 시점을 기준으로 구현한다.

: 너무나도 당연한 선지. 애초에 원금감과 입체감을 렌더링 과정에서 구현하는데 렌더링 과정은 관찰 시점을 기준으로 2차원 화면을 생선하는 단계이다. 적절.

④ 3D 영상을 재현하는 화면의 해상도가 높을수록 연산 양이 많아진다.

: 적절.

⑤ 병목 현상은 연산할 데이터의 양이 처리 능력을 초과할 때 발생한다.

: 병목 현상 생김->그러니까 그래픽 처리 능력을 보완 ; 처리 능력을 넘어설 만큼 데이터의 양이 많단 것. 적절.

35. 모델링 에 대한 설명으로 가장 적절한 것은?

모델링에 대해 묻는다. 단순히 모델링의 정보를 통해 풀기 이전에 모델링이 3D 영상에서 어떤

점을 특히 담당하는지, 그리고 당연히 병렬적으로 연결된 정보였던 렌더링과 구별할 수 있는 지를 묻는 것이다.

① 다른 물체에 가려져 보이지 않는 부분에 있는 삼각형의 정점들의 위치는 계산하지 않는다.
:모델링은 3차원의 영역을 다룬다. 특정 시점에서 가려진 부분/보이는 부분의 문제가 아니라 물체의 입체적 전 영역을 다루는 단계이다. 때문에 모든 부분을 다 챙겨야만 한다. 적절하지 않음.

② 삼각형들을 조합함으로써 물체의 복잡한 곡면을 정교하게 표현할 수 있다.
:시력검사인지 묻는 선지같기도 하다. 평가원의 자비로움에 박수를 친다. 적절.정답!

③ 하나의 작은 삼각형에 다양한 색상의 표면 특성들을 함께 부여한다.
:모델링에서 활용하는 각 삼각형들은 물체의 표면을 표현한다. 개별 삼각형들이 모여 전체 표면을 구현하므로 한 삼각형에 다양한 표면 특성을 부여하는 것이 아니라 하나의 삼각형에 하나의 표면 특성을 부여하고, 대신 삼각형을 다수로 쓰는 것이다. 적절하지 않음.

④ 공간상에 위치한 정점들을 2차원 평면에 존재하도록 배치한다.
:1번 선지와 같은 시각으로 접근한다. 모델링은 3차원의 영역!

⑤ 다양하게 변할 수 있는 관찰 시점을 순차적으로 저장한다.
:관찰 시점은 2차원 영역이다. 적절하지 않은 선지.

36. ㉠에 대한 추론으로 적절한 것은?

GPU에 대해 묻는다. 앞선 지문 해설에서 언급한 대로 CPU에 비해 보완된 점, 그리고 그것이 어떤 경우에 일어나는지가 핵심이다.

① 동일한 개수의 정점 위치를 연산할 때, 동시에 연산을 수행하는 코어의 개수가 많아지면 총 연산 시간이 길어진다.
:수행해야 할 연산의 수는 동일하고 코어의 개수만 변인으로 둔다. 코어의 개수가 늘어난다고 연산 시간이 길어지는가? 애초에 그러지 못하는데, 코어가 모든 데이터 연산을 동시에 하기 때문이다. 적절하지 않은 선지. (+시간이 길어질 것이면 왜 GPU가 굳이 여러 개의 코어를 가지겠는가...)

② 정점의 위치를 구하기 위한 10개의 연산을 10개의 코어에서 동시에 진행하려면, 10개의 연산 명령어가 필요하다.
:GPU의 핵심, 즉 장점은 하나의 연산 명령어로 다수의 연산을 동시에 하는 것이다. 핵심과 완벽히 어긋난 선지. 적절하지 않음.

③ 1개의 코어만 작동할 때, 정점의 위치를 구하기 위한 연산 시간은 1개의 코어를 가진 CPU의 연산 시간과 같다.
:GPU의 코어는 CPU의 코어보다 연산이 느리다. 그렇기에 대신 여러 개의 코어로 여러 데이터를 동시에 연산한다. 이 선지의 경우 GPU의 메리트가 완전히 삭제되었다. 그렇기에 CPU보다 저속 연산을 하여 시간은 더 길어진다. 적절하지 않음.

④ 정점 위치를 구하기 위한 각 데이터의 연산을 하나씩 순서대로 처리해야 한다면, 다수의 코어가 작동하는 경우 총 연산 시간은 1개의 코어만 작동하는 경우의 총 연산 시간과 같다.
:각 데이터의 연산을 하나씩 순서대로 처리해야 한다는 것에 주목하자. 문제가 묻는 핵심 중 하나인 '연산이 어떤 경우로 일어나는지'에 초점이 맞춰져 있다. 하나씩 순서대로 처리한

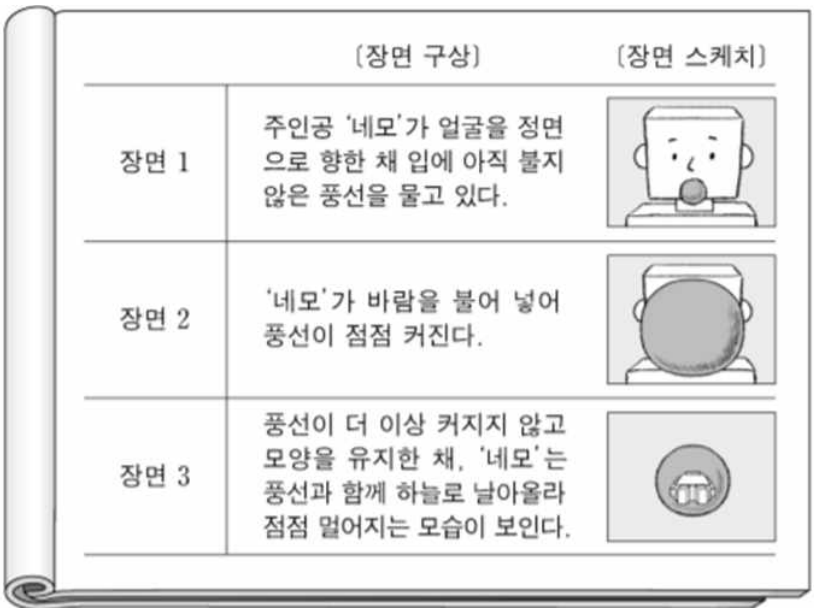
다면 GPU는 모든 메리트를 잃는다. 때문에 다수의 코어를 둔다한들 소용이 없어 1개의 코어로만 작동하는 경우와 연산 시간이 같을 수 밖에 없다. 적절한 선지. 정답!

⑤ 정점 위치를 구하기 위해 연산해야 할 10개의 데이터를 10개의 코어에서 처리할 경우, 모든 데이터를 모든 코어에 전송하는 시간은 1개의 데이터를 1개의 코어에 전송하는 시간과 같다.

:지금 이 선지는 '데이터를 코어에 전송하는 시간'에 대해 묻는다. 연산 시간이 아니다!

GPU는 데이터를 순차적으로 코어에 전송하기 때문에 데이터가 많을수록, 또 코어가 많을수록 전송시간은 길 수 밖에 없다. 간단한 선지인데 은근 함정을 가지고 있는 선지. 적절하지 않음.

37. 다음은 3D 애니메이션 제작을 위한 계획의 일부이다. 윗글을 바탕으로 할 때 적절하지 않은 것은?



주어진 세 장면을 어떻게 구현할지에 대해 묻는 문제이다. 당연히 우리가 학습한 모델링과 렌더링을 이용할 것이며, 각 단계에서 어떤 것이 핵심인지를 장면을 보고 예상할 수 있어야 한다.

장면1의 경우 정지된 화면이다. 때문에 모델링에서 물체를 어떻게 구현하는지, 또 렌더링에선 어떤 시점으로 그 물체를 바라볼지가 핵심.

장면2의 경우 풍선의 크기가 점점 커지는 것에 초점을 둔다. 즉 물체의 크기가 커지는 것이므로 모델링에서 정점 사이의 간격이 넓어지는 것이 핵심.

장면3의 경우 물체의 모양이나 크기 자체는 유지되나 화면 안에서 멀어지게 된다. 즉 렌더링 단계에서 원근감을 구현하고 하는 것이 핵심. 화솟값 지정의 문제다.

① 장면 1의 렌더링 단계에서 풍선에 가려 보이지 않는 입 부분의 삼각형들의 표면 특성은 화솟값을 구하는 데 사용되지 않겠군.
:렌더링은 2차원 화면에서 특정 관찰 시점을 기준으로 입체적 물체를 바라보는 것을 구현한다. 즉 우리가 바라보는 그 장면에서 보이는 것들만 신경을 쓰면 된다. 때문에 풍선에 가려 보이지 않는 부분은 전혀 신경쓸 필요가 없다. 적절한 선지.
+)복수정답의 여지 존재하지 않음. 렌더링 단계는 물체를 바라보는 관찰 시점이 절대적 기준이므로 관찰 시점에서 보이지 않는 부분은 고려하지 않음.

② 장면 2의 모델링 단계에서 풍선에 있는 정점의 개수는 유지되겠군.
:역시 위에 언급한 핵심을 피해가지 않는다. 물체인 풍선은 크기가 커질 뿐이므로 정점 사이의 간격만 넓어지며 정점의 개수는 변하지 않는다. 애초에 정점들의 개수는 물체가 변형되어도 변하지 않는다! 적절한 선지.
③ 장면 2의 모델링 단계에서 풍선에 있는 정점 사이의 거리가 멀어지겠군.
:핵심에 들어맞는다. 적절한 선지.

④ 장면 3의 모델링 단계에서 풍선에 있는 정점들이 이루는 삼각형들이 작아지겠군.
:장면3은 물체의 절대적 변형이 아니다. 물체는 유지가 되면서 화면 상에서 멀어지는 것이다. 때문에 모델링 단계에서 특정한 변화가 일어나지 않는다. 그저 렌더링 단계에서의 화솟값의 문제이다. 적절하지 않은 선지. 정답!
⑤ 장면 3의 렌더링 단계에서 전체 화면에서 화솟값이 부여되는 화소의 개수는 변하지 않겠군.
: 전체 화면에서 화솟값이 부여되는 화소의 개수는 화솟값을 부여하기 전에 정해진다. 즉 화면상 물체가 변화가 생긴다고 해도 이미 정해진 화소의 개수는 변하지 않는다. 적절한 선지.