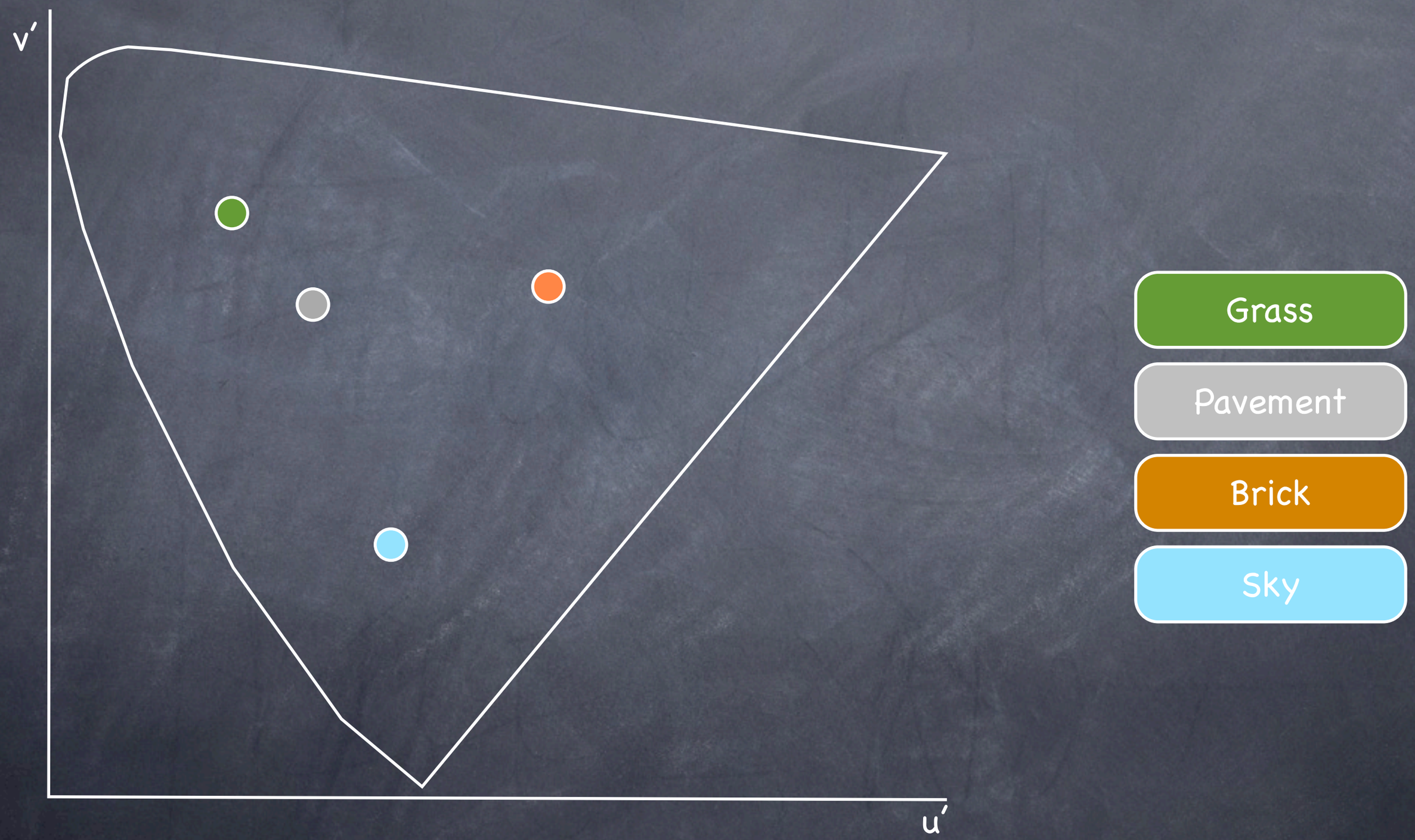
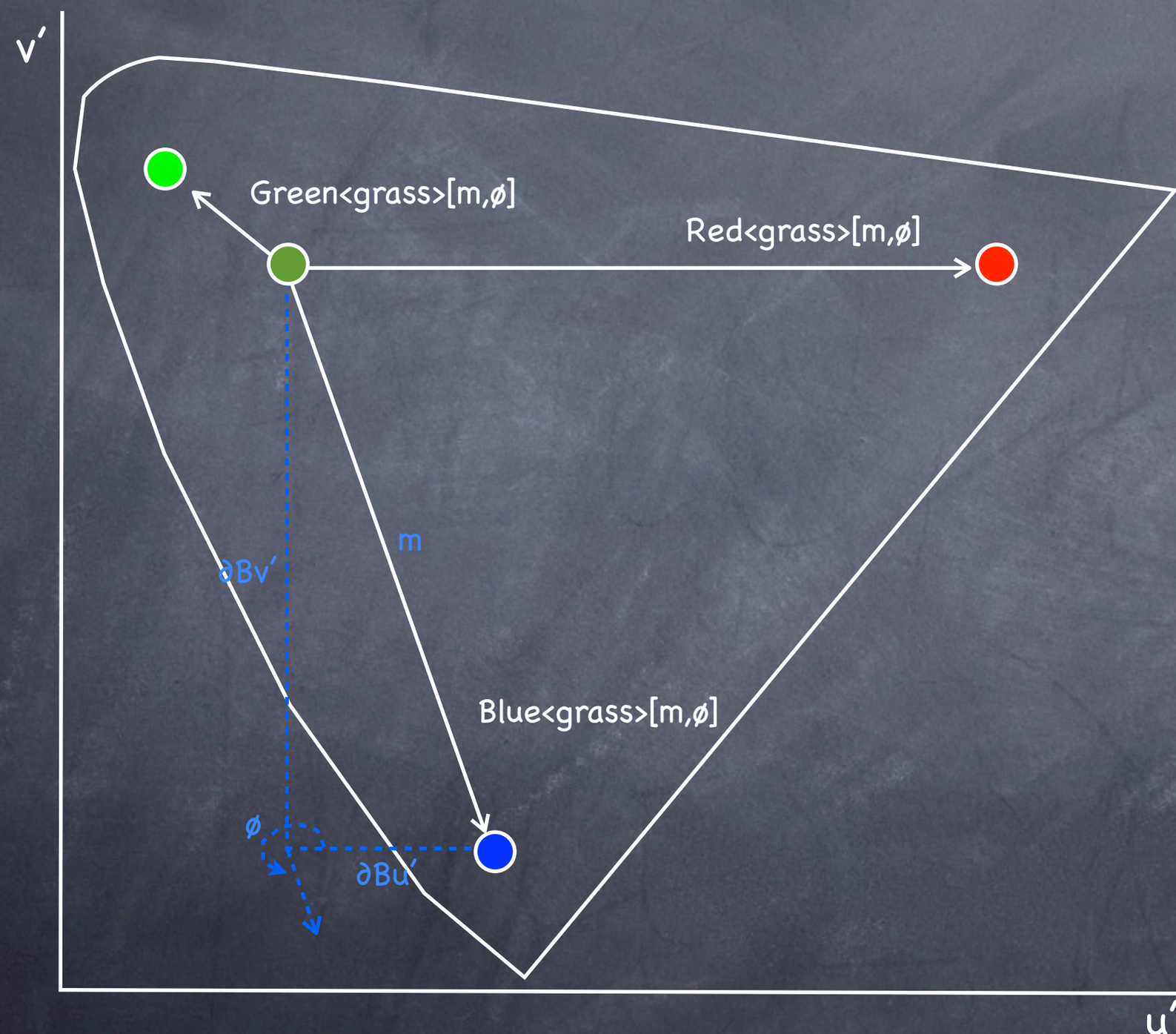
Color Correction on See-Through Displays

Fundamentals and Experimental Approach

Fundamentals -- Four backgrounds



For a given BG there are three vectors to R,G,B



Grass

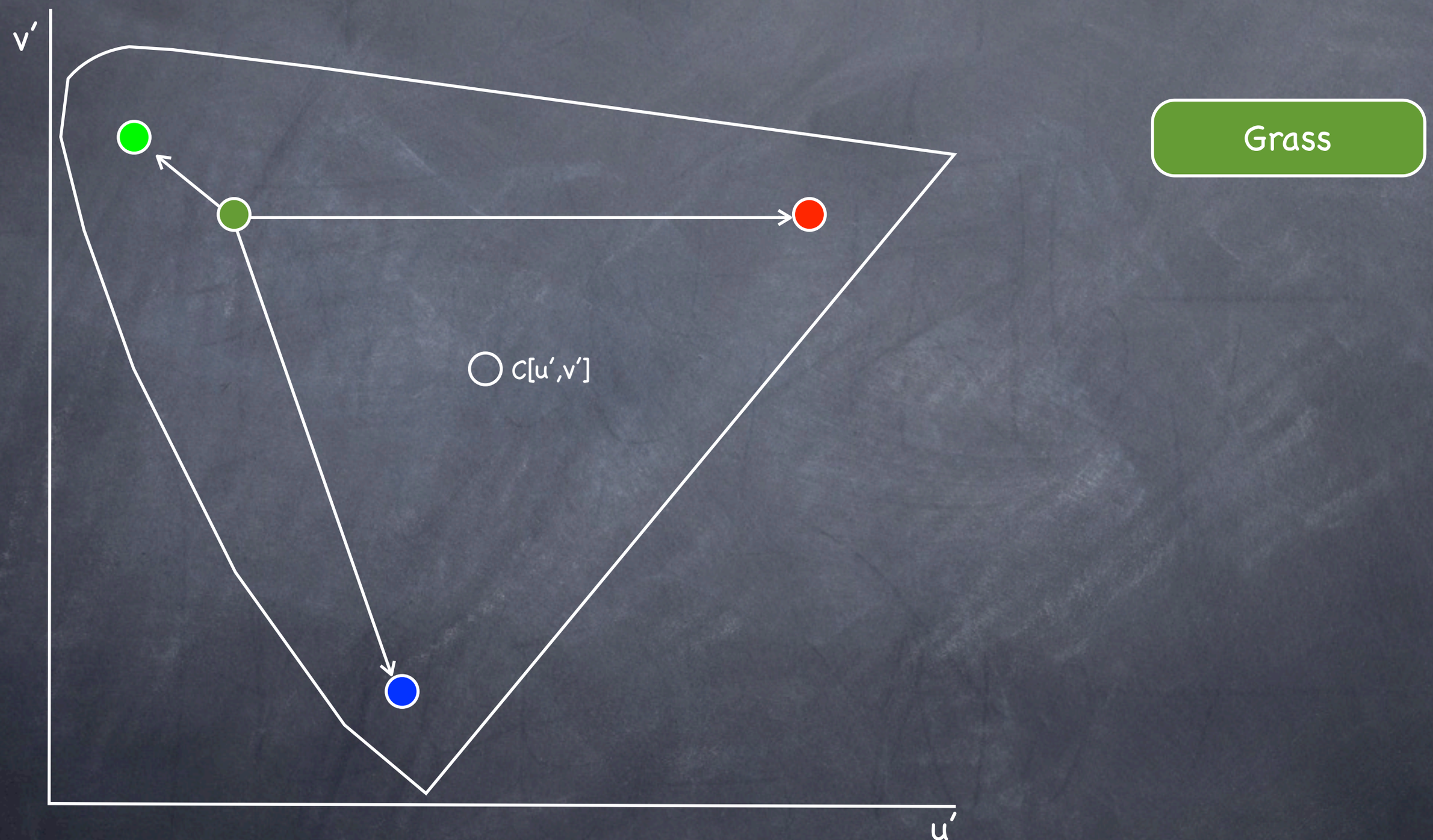
Grass = [GrassU', GrassV']

PrimeVectors<grass> =
[
Red<grass>[m,ø],
Green<grass>[m,ø],
Blue<grass>[m,ø]
]
=
[
[∂Ru', ∂Rv'],
[∂Gu', ∂Gv'],
[∂Bu', ∂Bv']
]

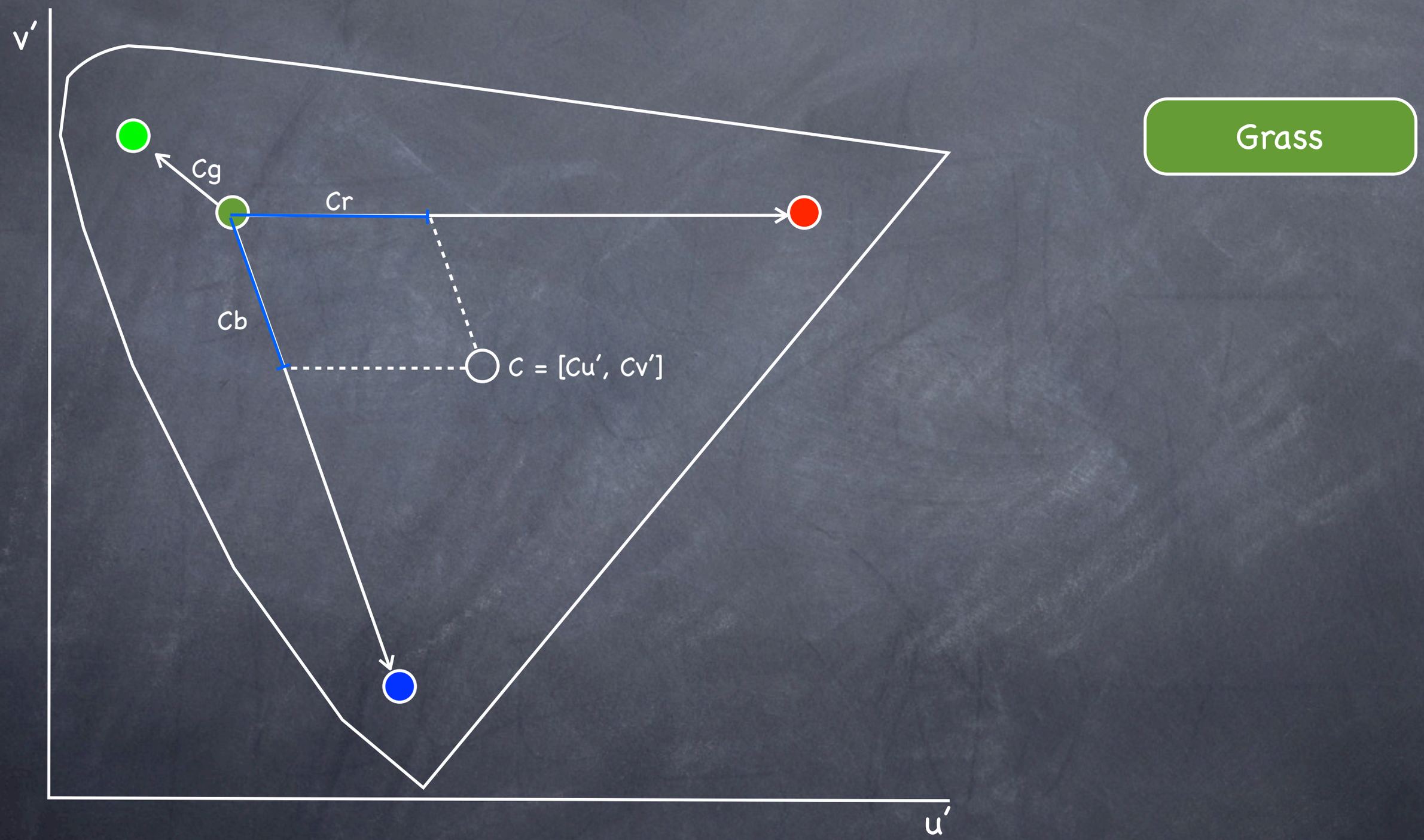
∂Bu' = |GrassU' - BlueU'|
∂Bv' = |GrassV' - BlueV'|

Same for Red and Green.

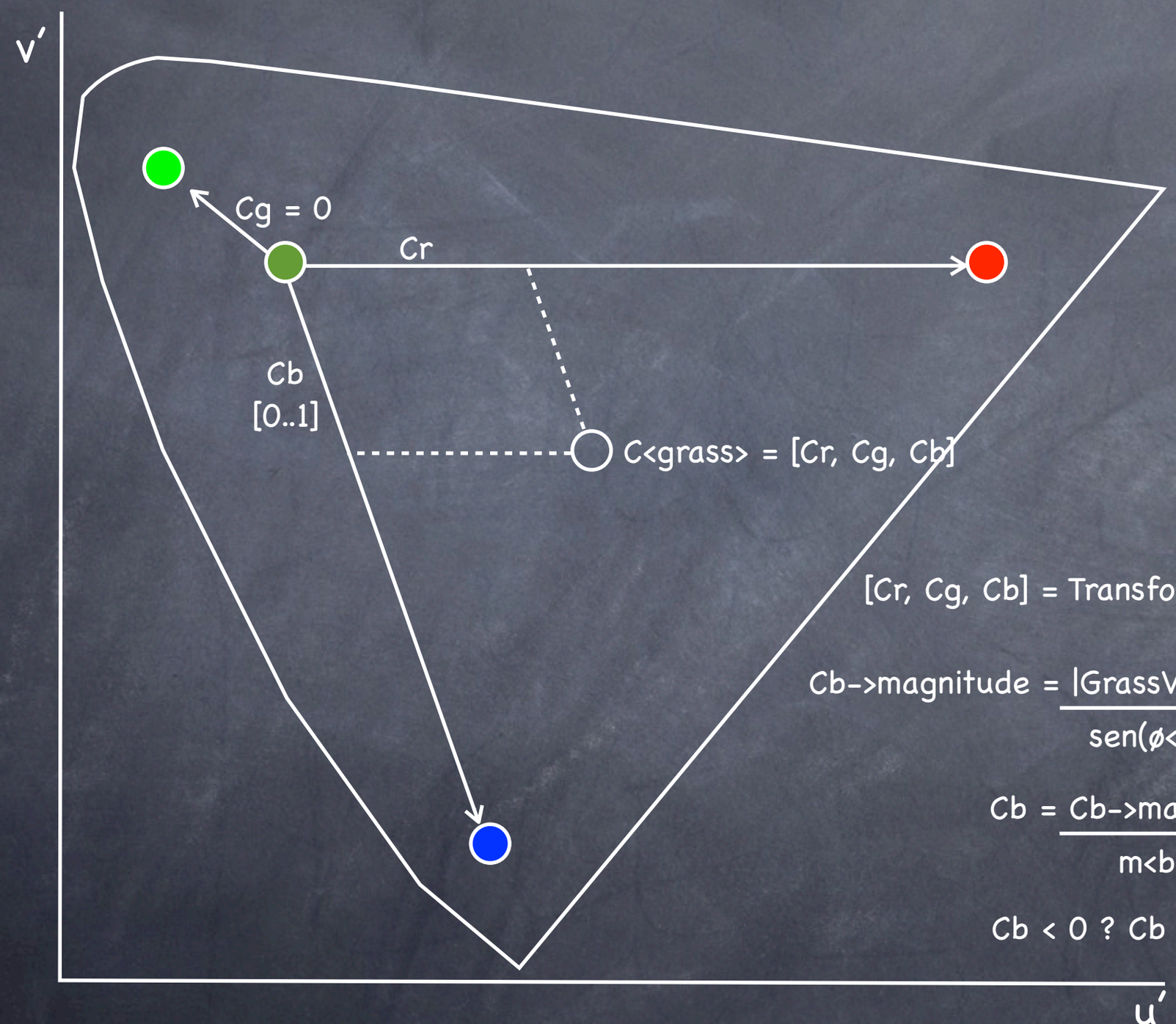
Any RGB color can be mapped to the U'V' space



Can also be expressed in terms of those vector



From C[u′,v′] to C<grass>[Cr, Cg, Cb]



Grass

[Cr, Cg, Cb] = Transform(Cu′, Cv′, Grass, PrimeVectors<grass>)

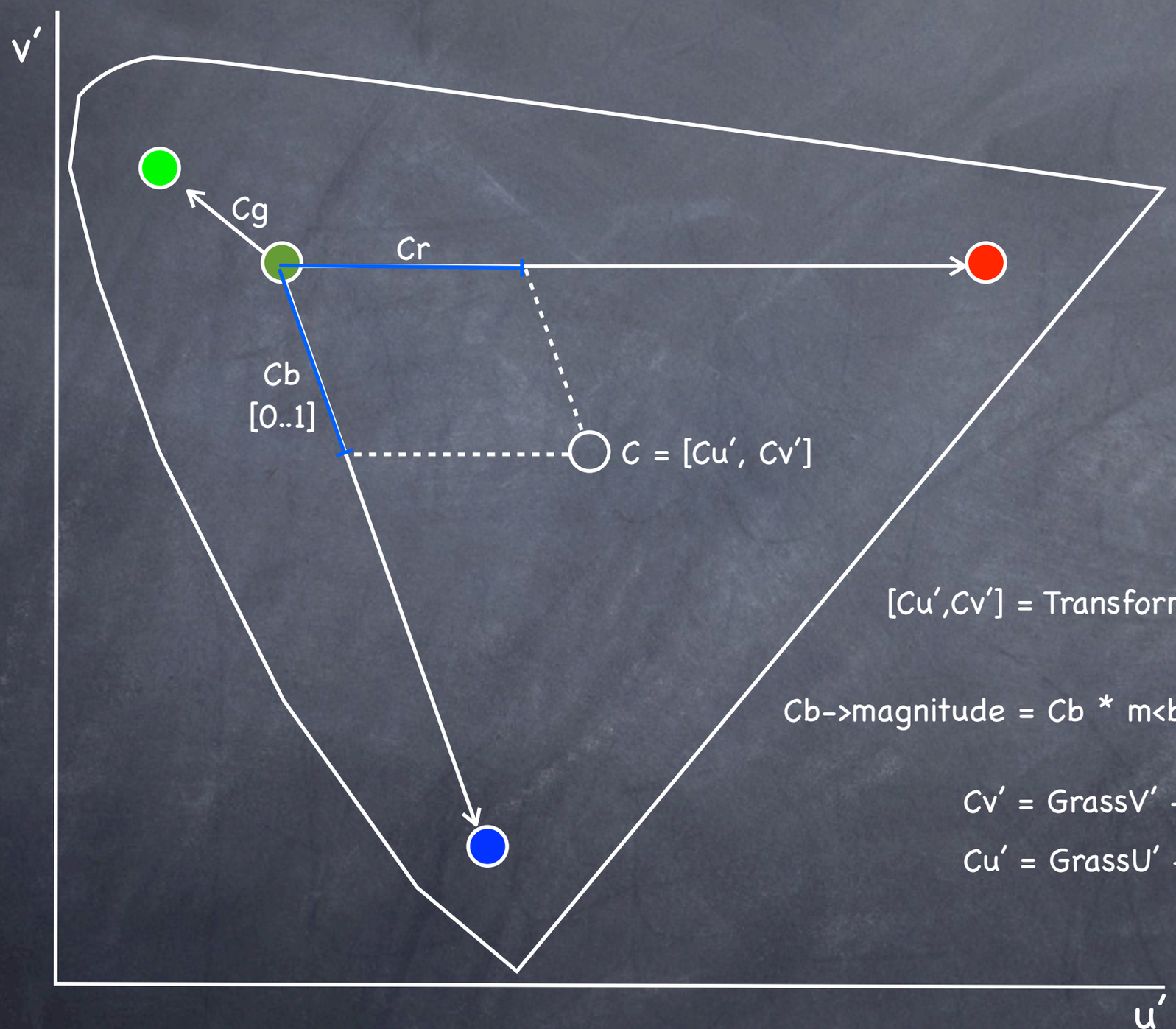
$$Cb\text{->}magnitude = \frac{|GrassV' - Cv'|}{sen(\emptyset\text{<blue>})}$$

$$Cb = \frac{Cb\text{->}magnitude}{m\text{<blue>}}$$

Cb < 0 ? Cb = 0

Same for Red and Green.

From C<grass>[Cr, Cg, Cb] to C[u′, v′]



Grass

[Cu′,Cv′] = Transform(Cr, Cg, Cb, Grass, PrimerVector<Grass>)

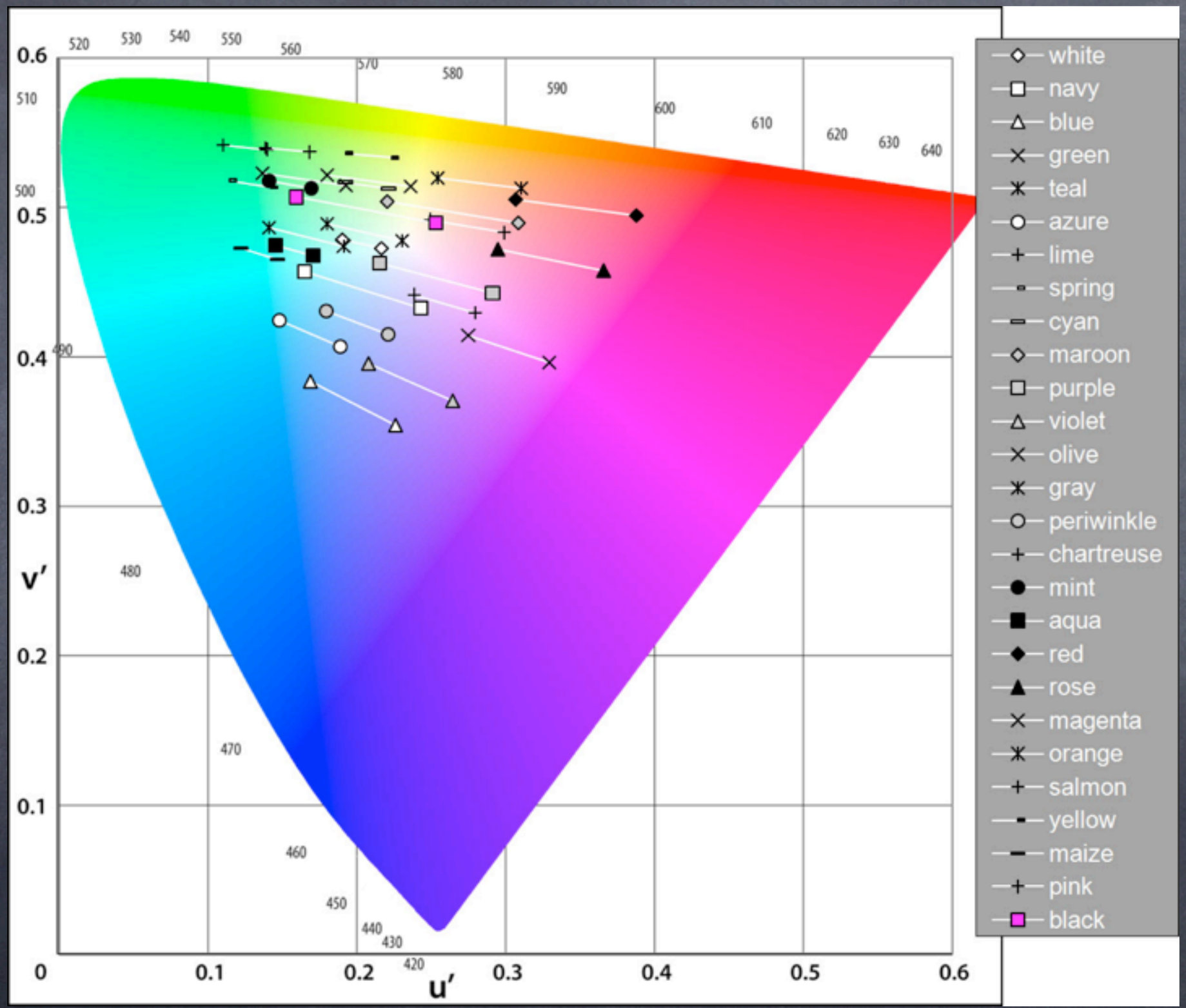
Cb->magnitude = Cb * m<blue>

Same for Red and Green.

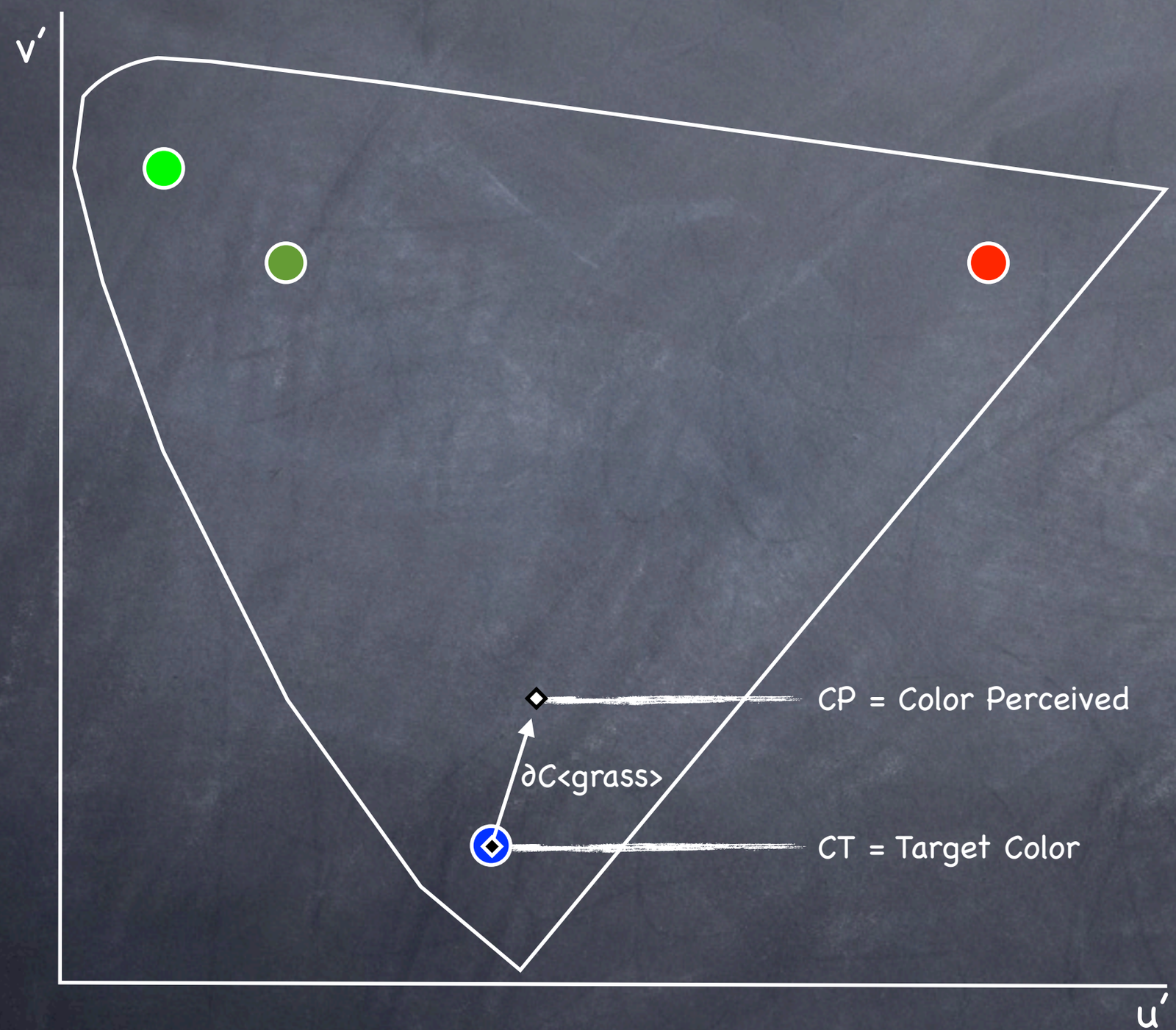
Cv′ = GrassV′ + Σ (Ci->magnitude * sen(ø<i>))

Cu′ = GrassU′ + Σ (Ci->magnitude * cos(ø<i>))

Color shift problem in AR



Shift vector



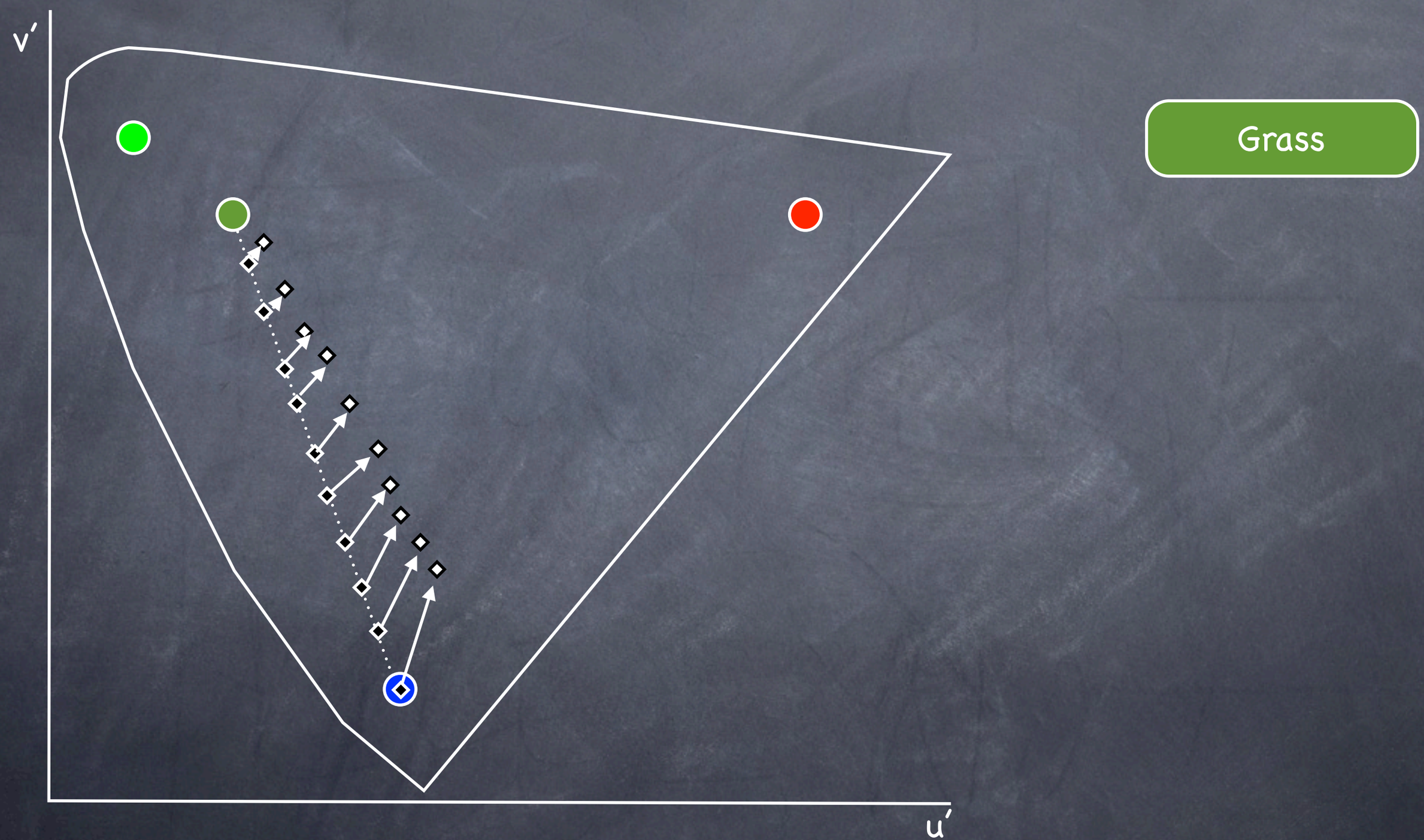
Grass

∂C<grass> = [m, ø]

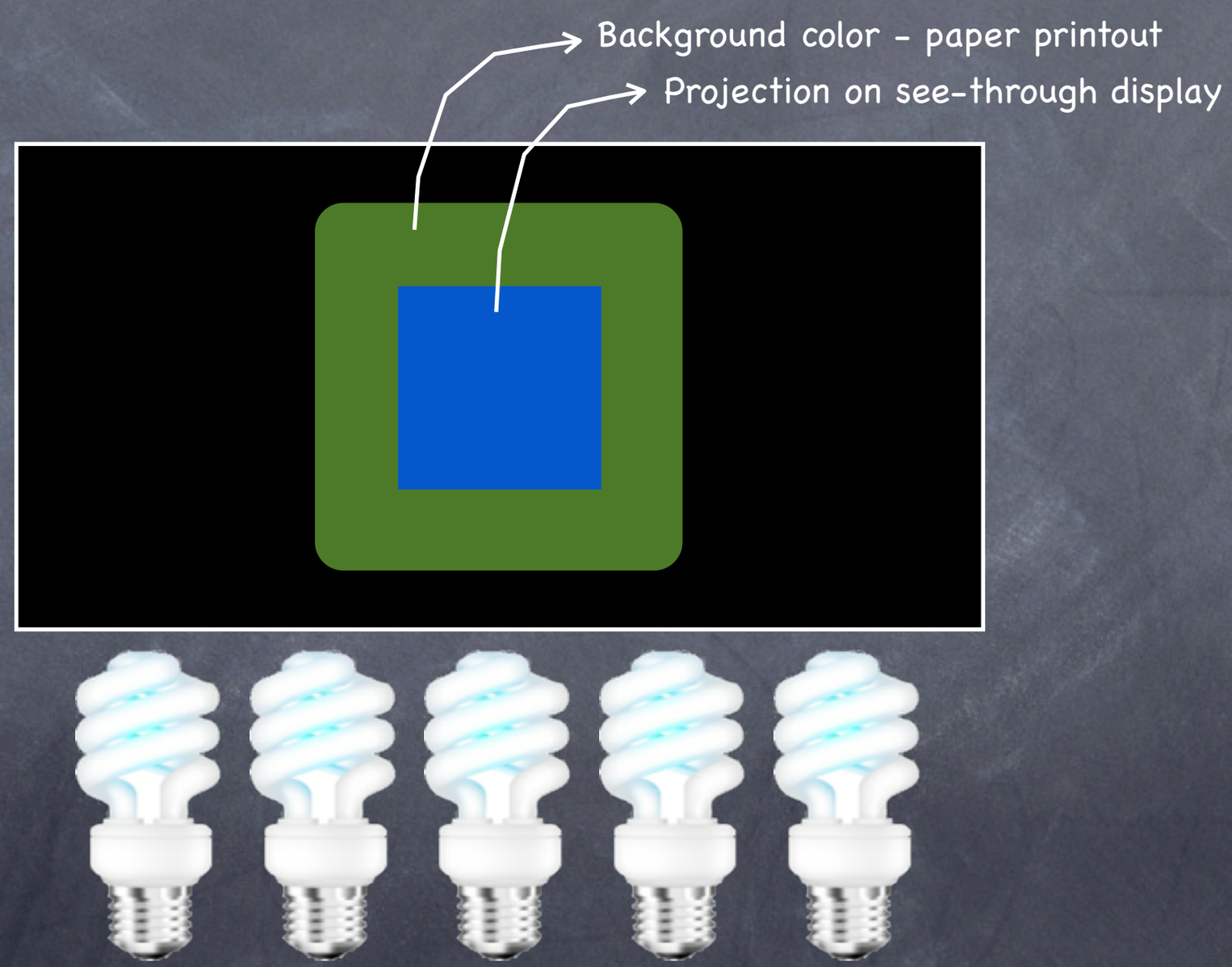
Research Questions?

- Can the color shift along a PrimeVector for a given background be characterized by a function?
- Can we use this function to estimate how any color will be perceived for a given background?
- Based on this functions for a given background, can we propose a color correction algorithm to determine a display color which will be perceived closer to the intended color.

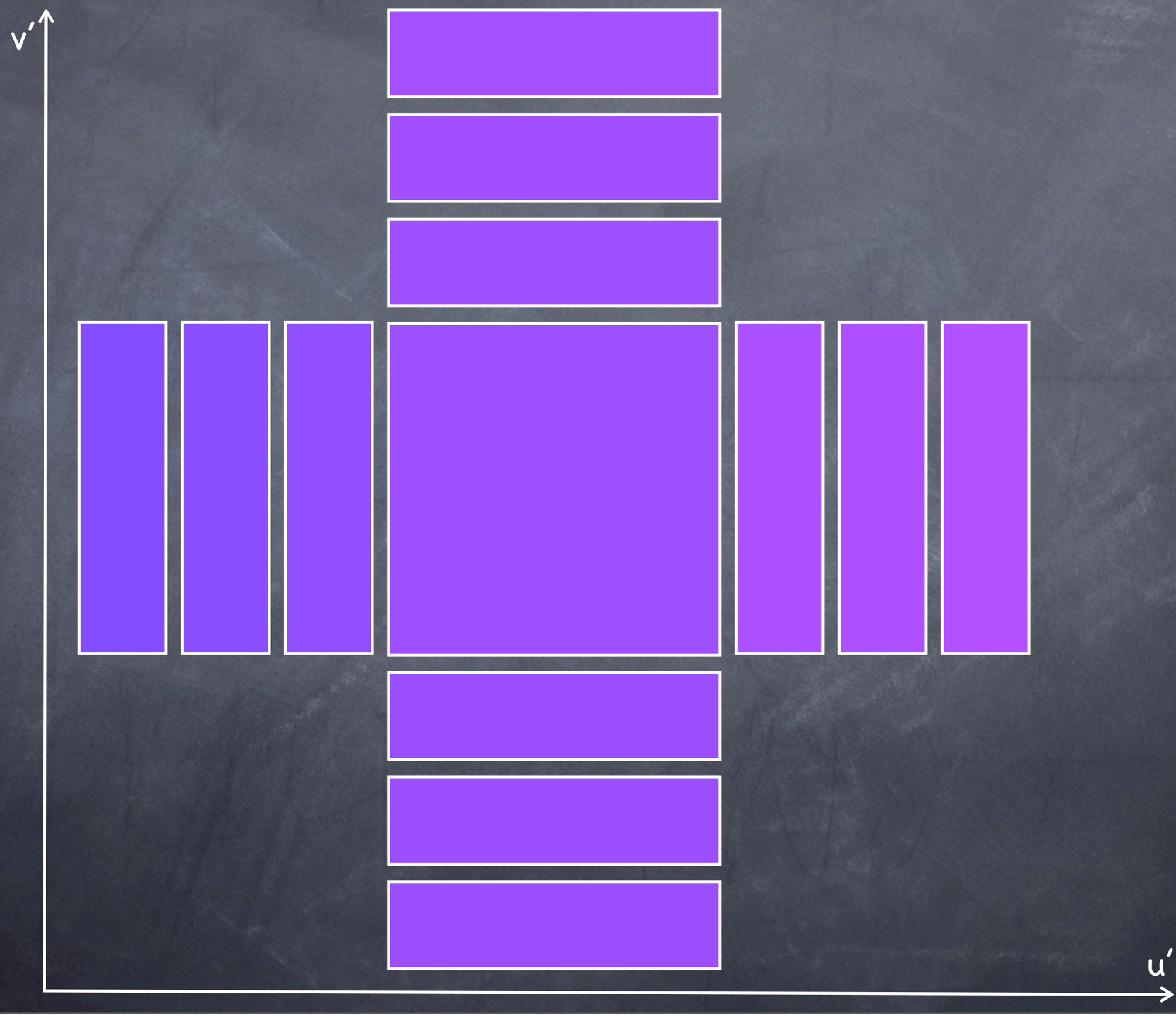
(1) - Shift along the axis of each PrimerVector



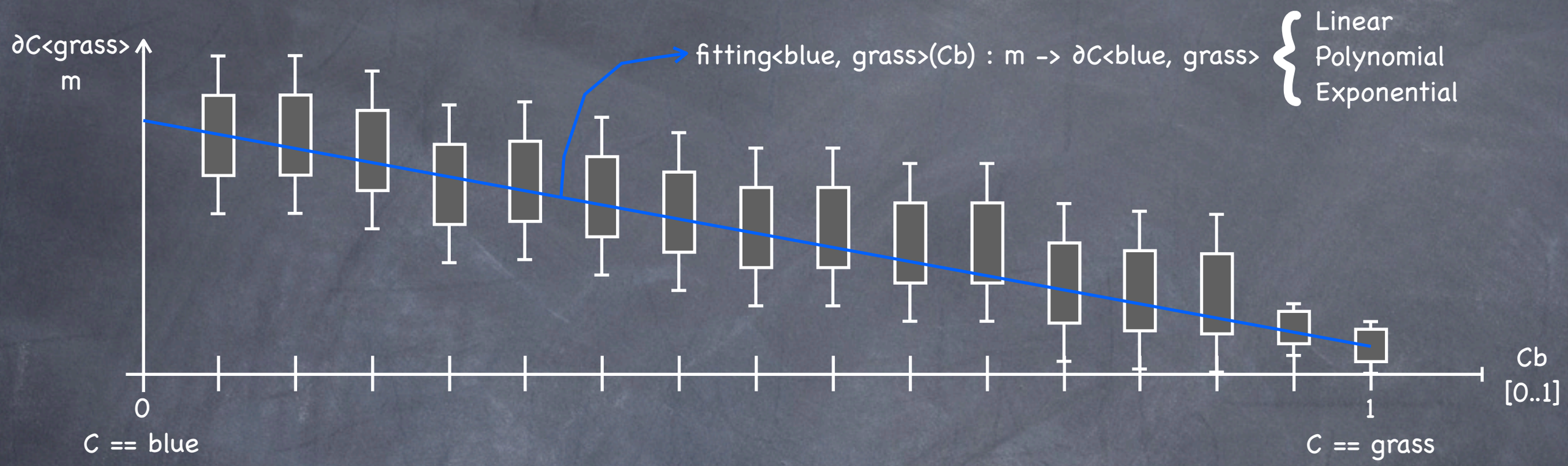
Hardware set-up for color-shifting projection

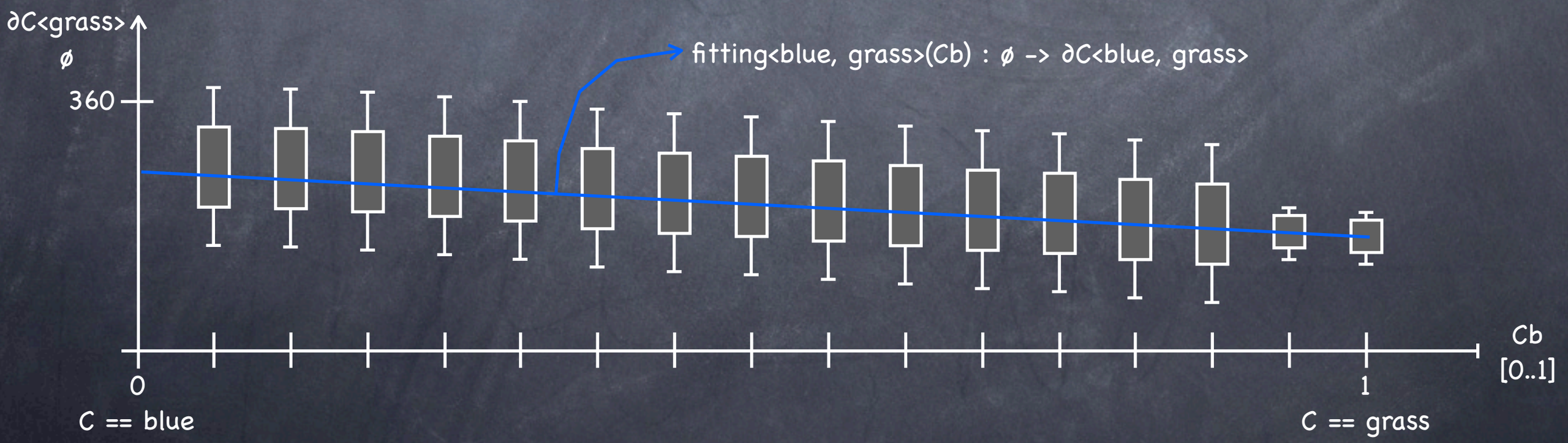


User UI to input the perceived color

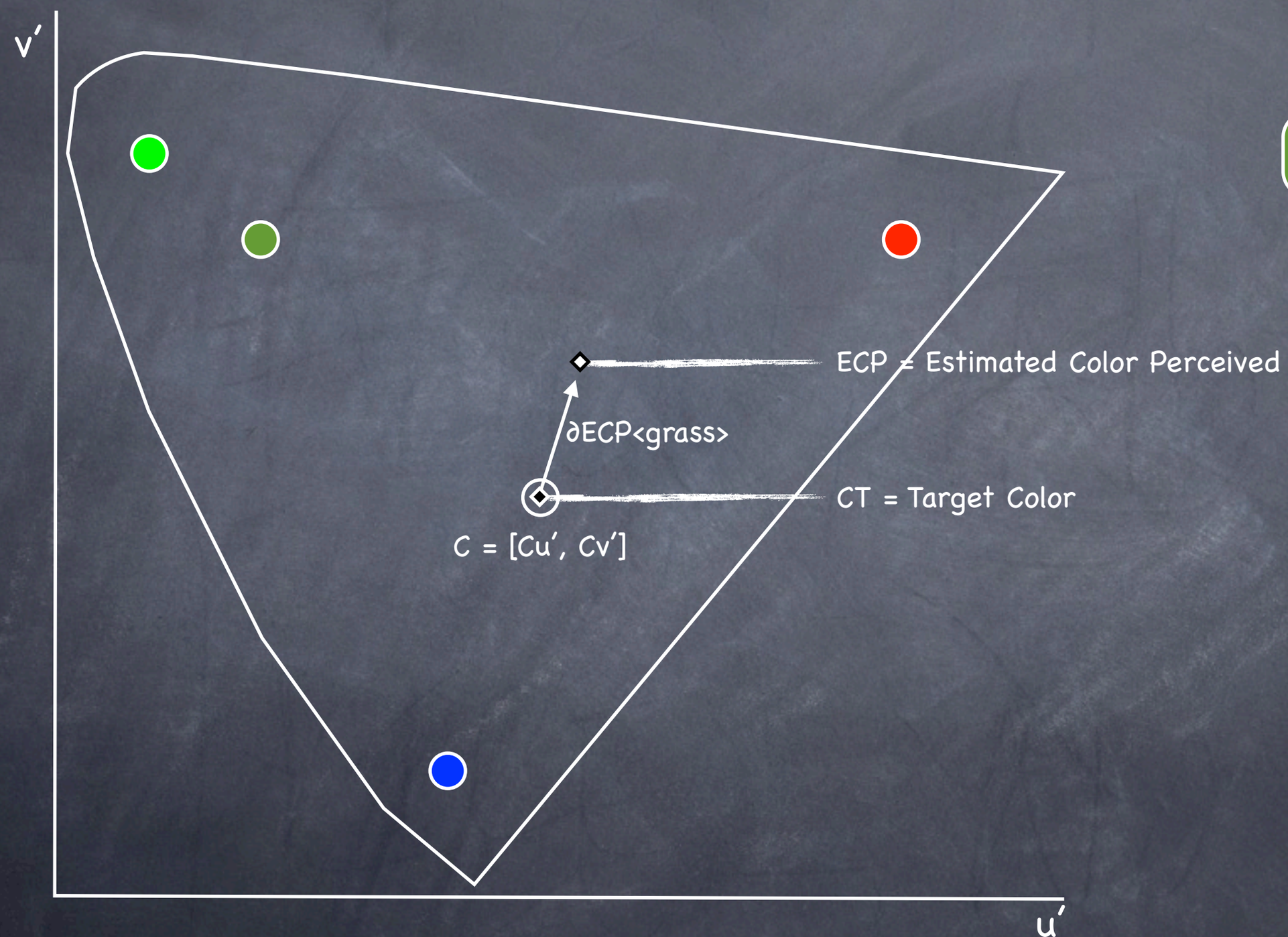


Find the shift function for each Component to BG





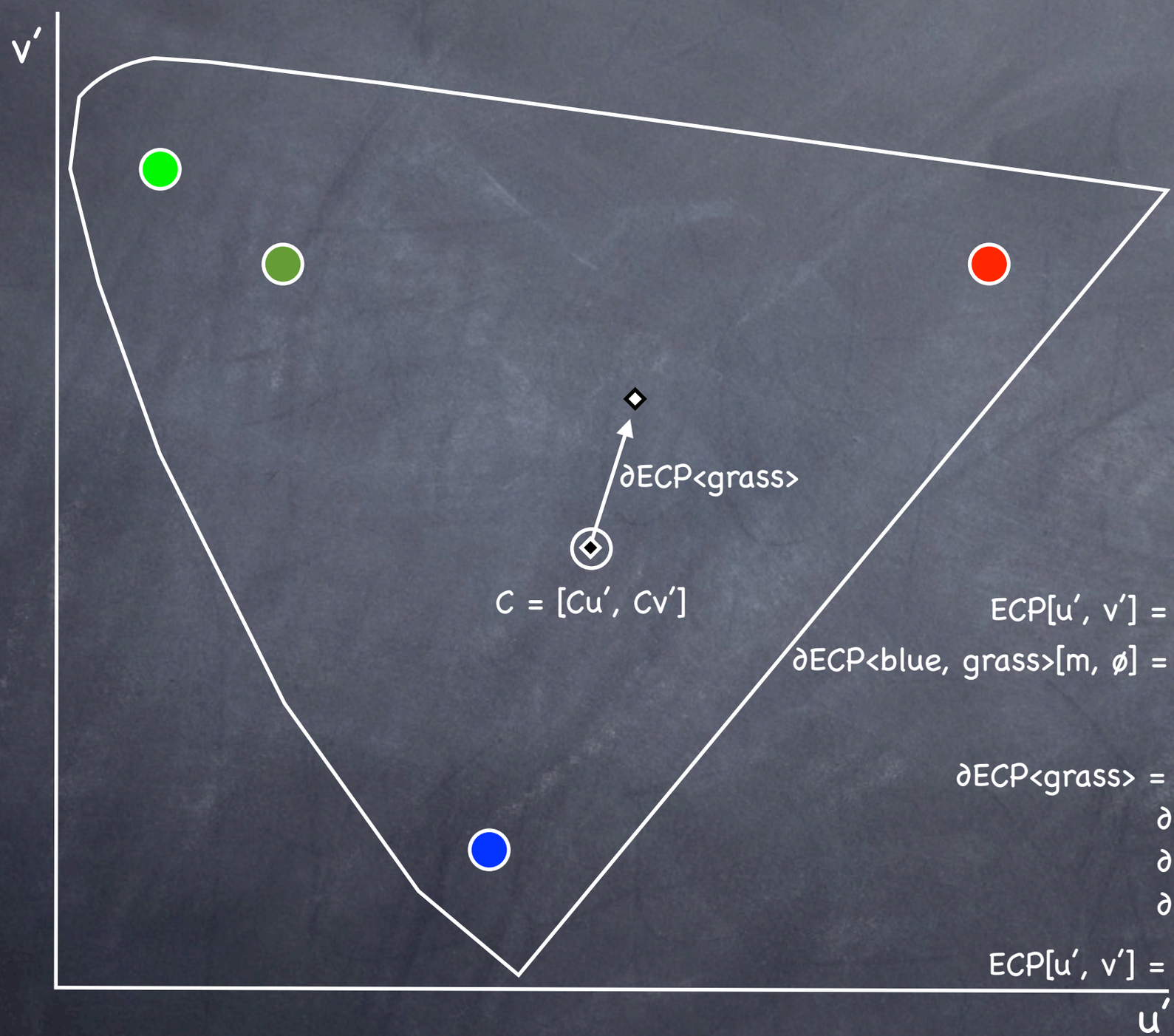
(2) - For any C[u′, v′] estimate the shift and evaluate the estimation



Grass

I can evaluate this against the colors used in the "more than meet the eye paper"

(2) - For any C[u′, v′] estimate the shift and evaluate the estimation



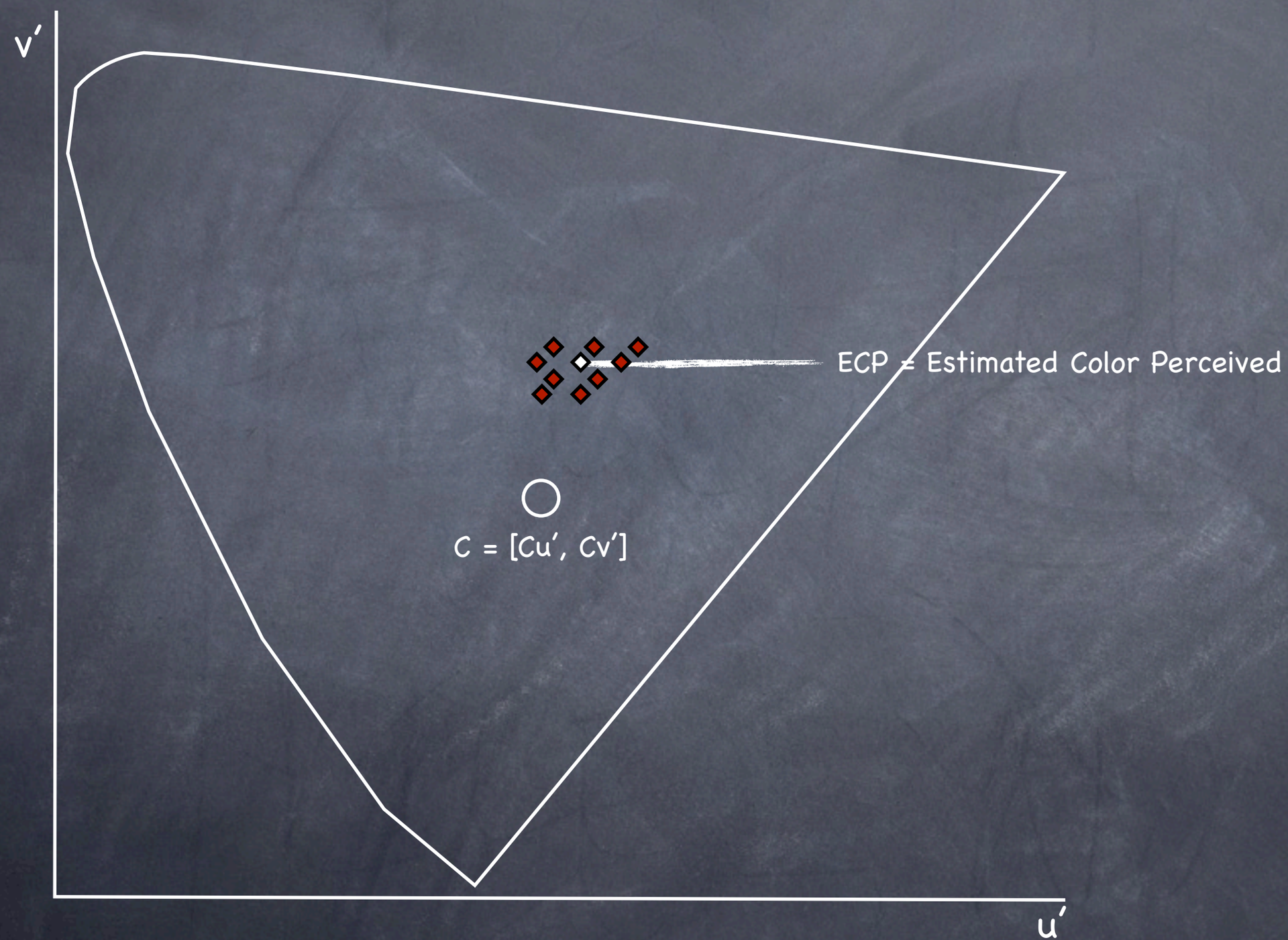
Grass

ECP[u′, v′] = Project(Cu′, Cv′, Grass) ⟶ [Cr, Cg, Cb]

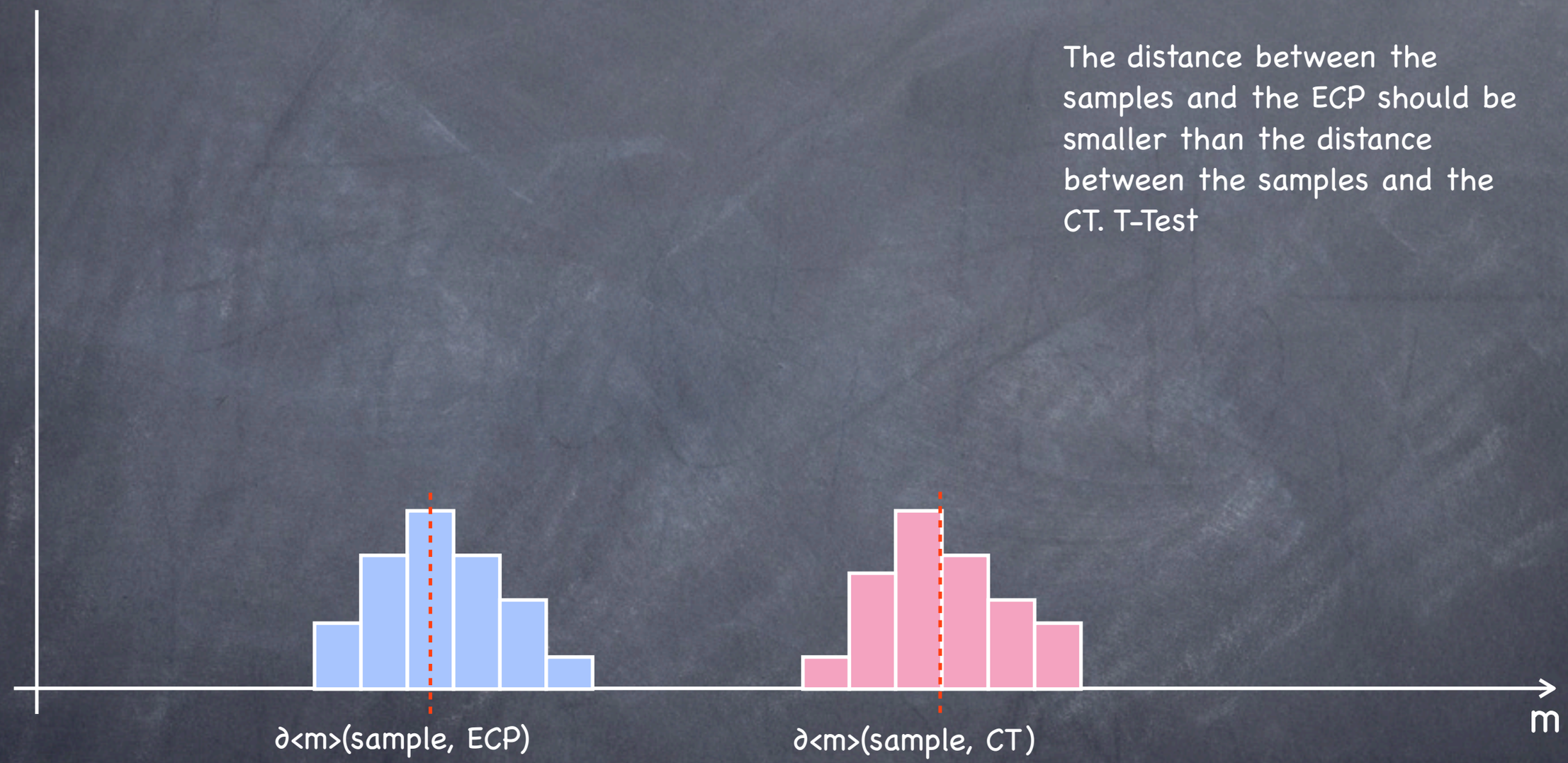
∂ECP<blue, grass>[m, ∅] =
fitting<blue, grass>(Cb) : m -> ∂C<grass>,
fitting<blue, grass>(Cb) : ∅ -> dC<grass>

∂ECP<grass> =
∂ECP<red, grass> +
∂ECP<green, grass> +
∂ECP<blue, grass>

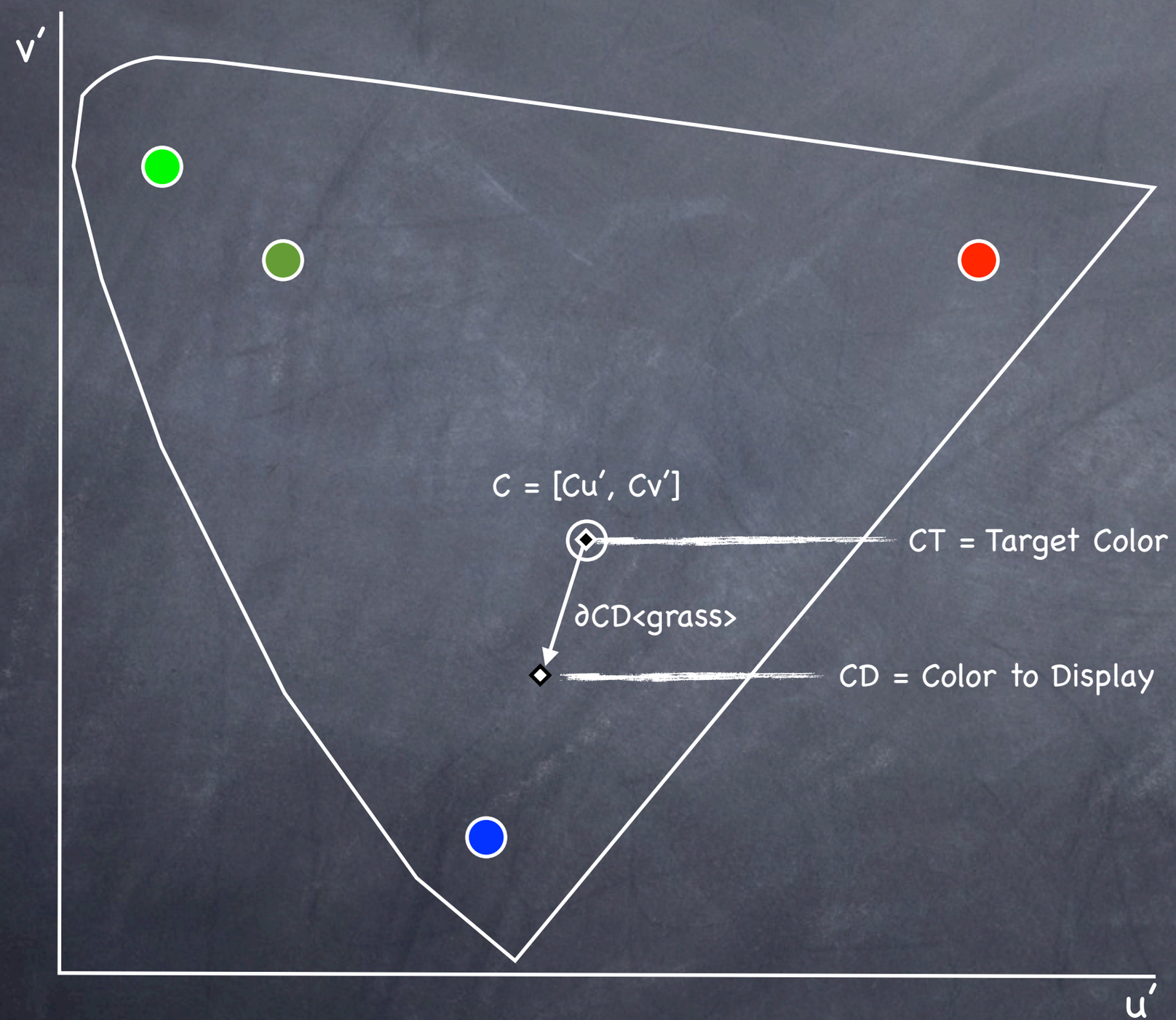
ECP[u′, v′] = [Cu′, Cv′] + ∂ECP<grass>[u′, v′]


v′
ECP = Estimated Color Perceived
C = [Cu′, Cv′]
u′

The distance between the samples and the ECP should be smaller than the distance between the samples and the CT. T-Test

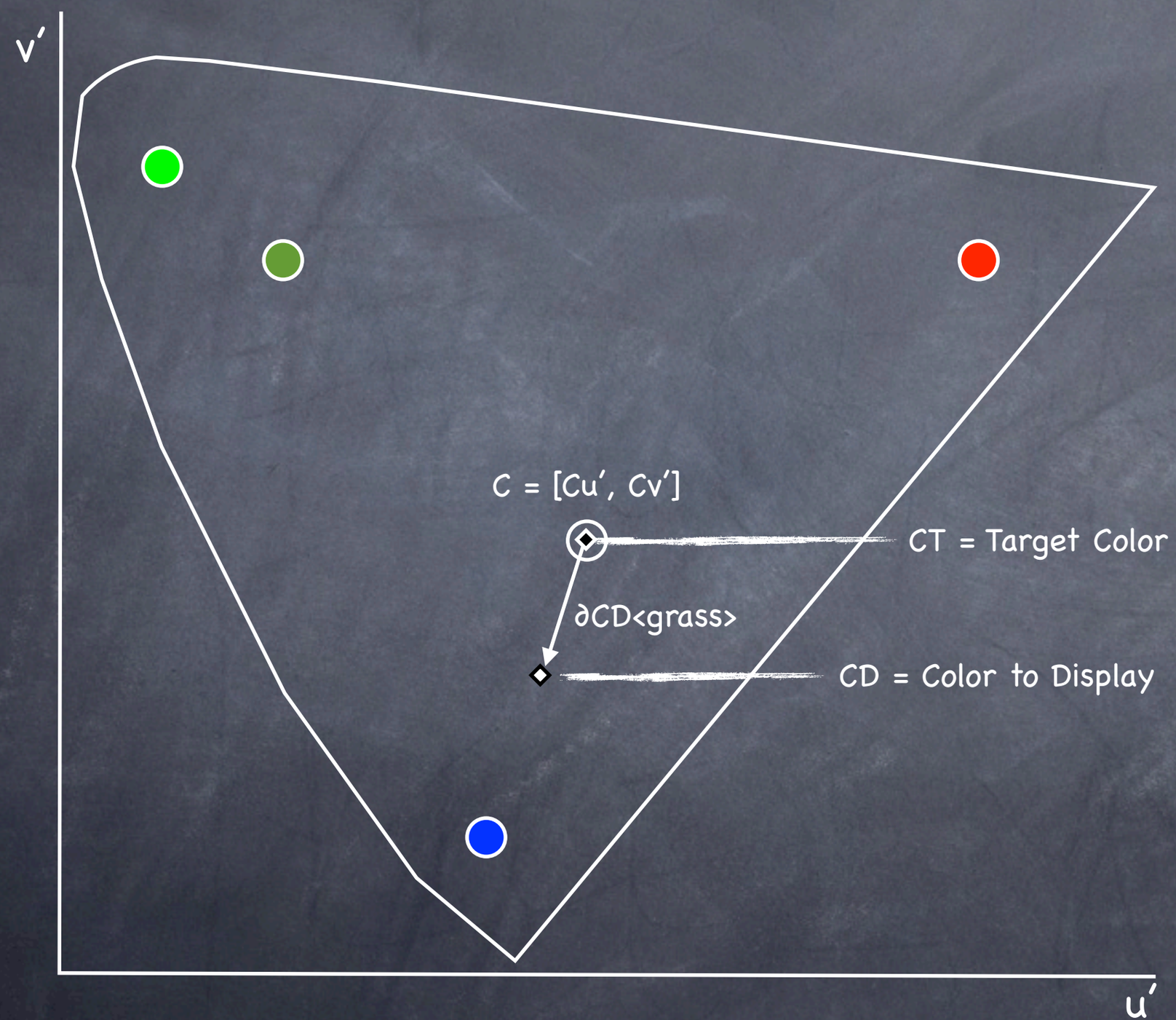


(3) - Find color-correction algorithm

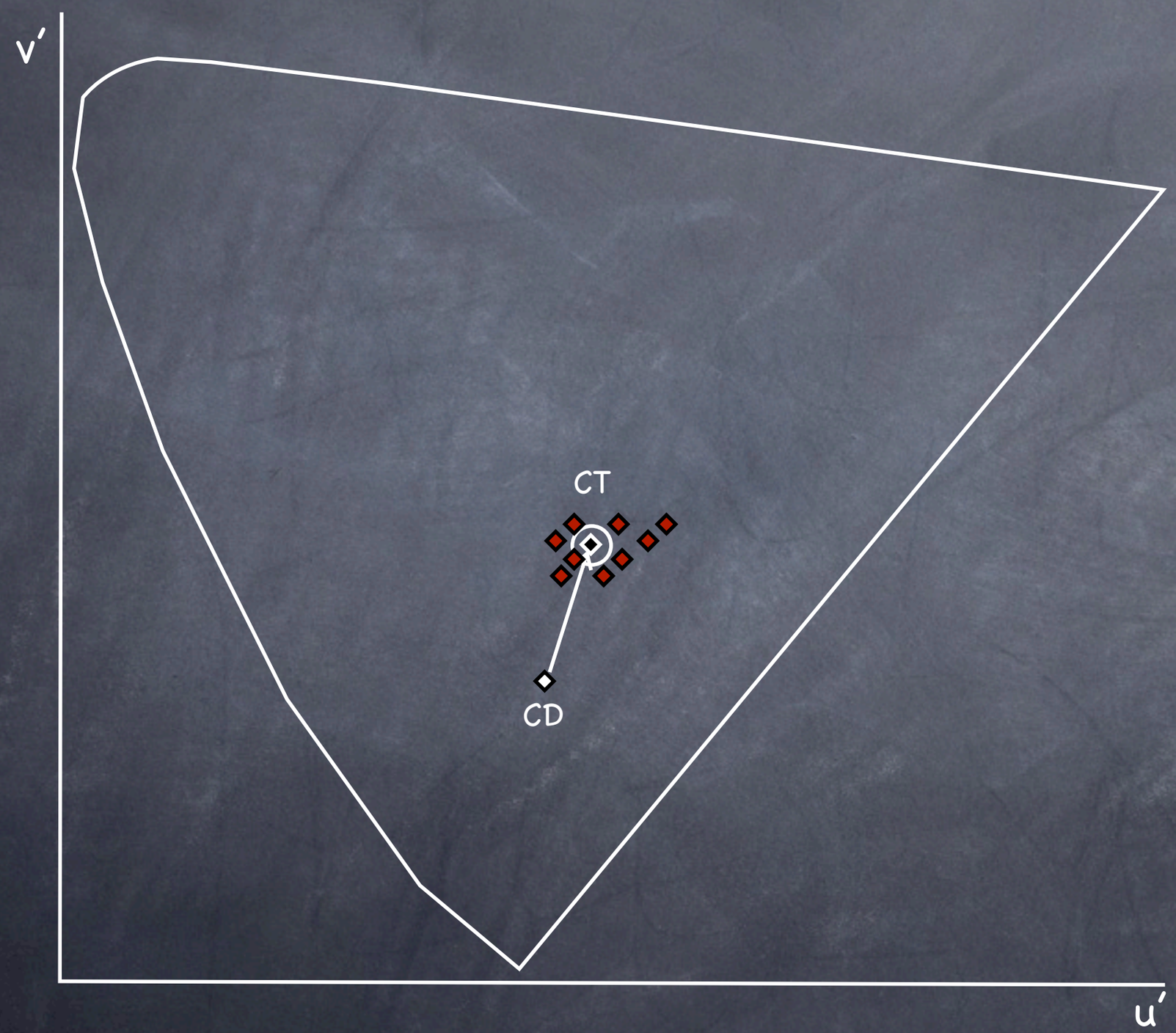


I can evaluate this against the colors used in the "more than meet the eye paper"

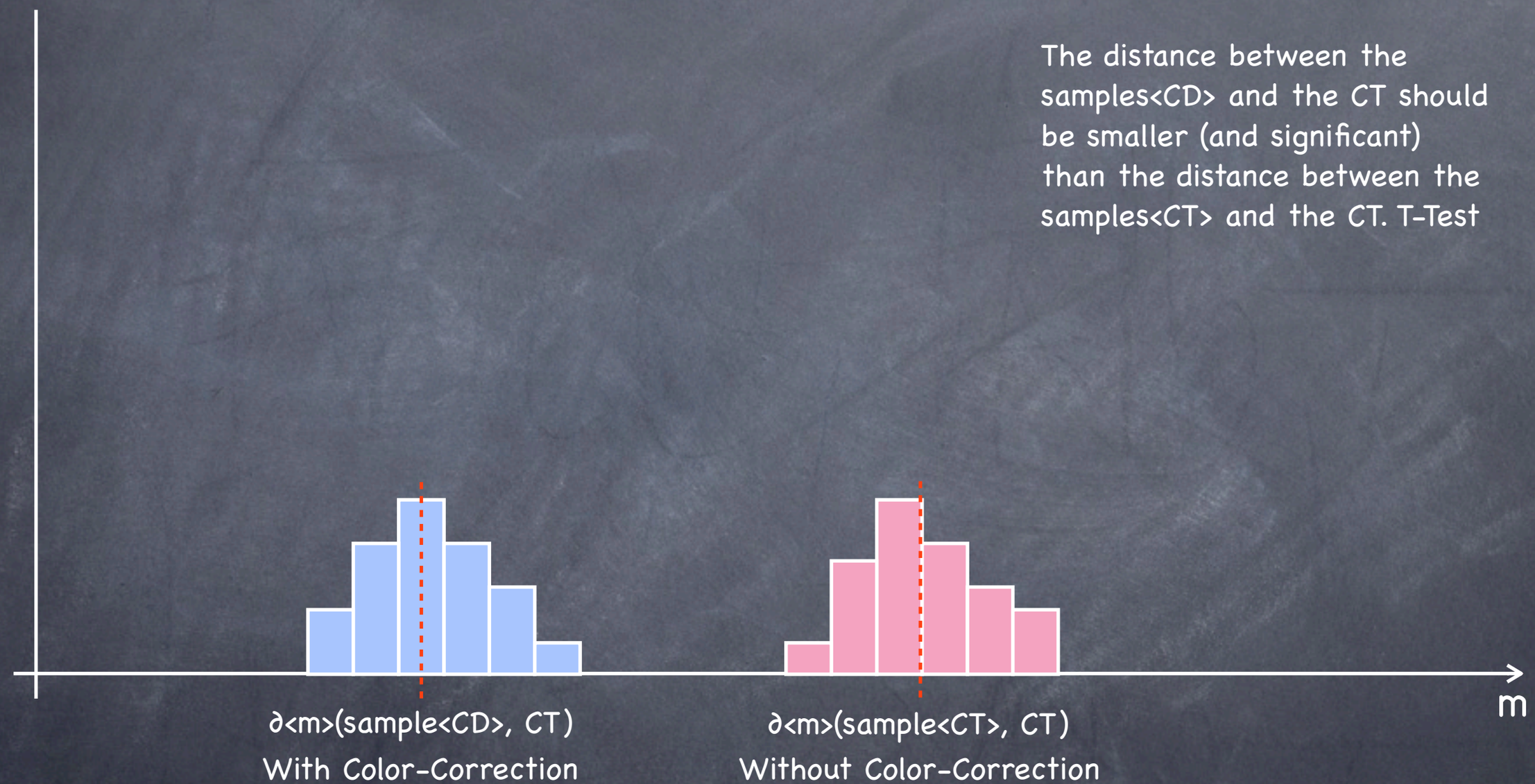
(3) - Find color-correction algorithm



Solve by using an approximation method where different CD are tested [10 tests] choosing the one that produces a ECP closer to CT.


v′
CT
CD
u′

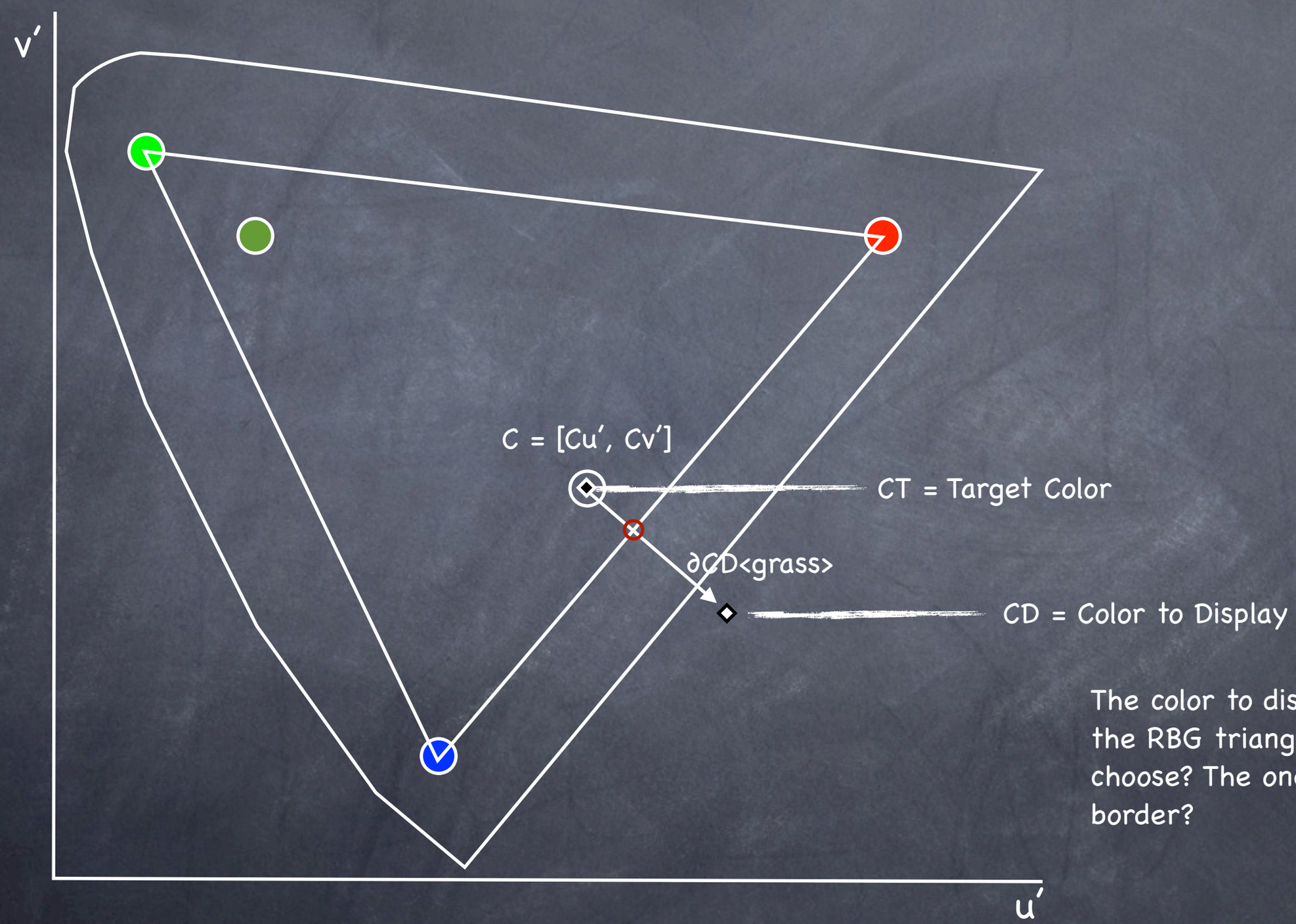
The distance between the samples<CD> and the CT should be smaller (and significant) than the distance between the samples<CT> and the CT. T-Test



Open Questions

- Initial position of the user assessment tool in the [u′, v′] space relative to the CT and the ECP.
- What's the most appropriate approximation method for determining the CD?
- How to handle the correction in border areas?

Color-Correction on a Border Area



The color to display falls out of the RBG triangle. What color to choose? The one right on the border?