

# Prokudin-Gorskii Görüntülerinin Otomatik Hizalanması ve Renk Restorasyonu

Yiğit Buğra KÜÇÜK  
Yapay Zeka Mühendisliği • 230212048  
230212048@ostimteknik.edu.tr • October 18, 2025

**Abstract**—Bu çalışma, Sergei Mikhailovich Prokudin-Gorskii'nin 20. yüzyıl başlarında geliştirdiği tarihi renkli fotoğrafçılık tekniğinin dijital restorasyonunu amaçlamaktadır. Özgün yöntemde üç ayrı cam plakaya kaydedilen RGB kanalları, Normalize Çapraz Korelasyon (NCC) metrik kullanılarak otomatik olarak hizalanmakta ve birleştirilmektedir. Geliştirilen algoritma 29 tarihi görüntü üzerinde test edilmiş ve %98.33 doğruluk oranında başarılı sonuçlar elde edilmiştir. Çalışmada brute-force arama, kenar kırpması ve çoklu görüntü iyileştirme teknikleri entegre edilmiştir.

## I. GİRİŞ

### A. Problemin Tanımı

Prokudin-Gorskii tekniği, her sahneyi sırasıyla mavi, yeşil ve kırmızı filtreler kullanarak üç ayrı cam plakaya kaydetmektedir. Dijital restorasyon sürecinde temel zorluk, bu üç kanal arasındaki geometrik kaymaların doğru şekilde tespit edilmesi ve renk dengesizliklerinin giderilmesidir. Bu çalışmada, tam otomatik bir kanal hizalama ve renk restorasyon pipeline'ı geliştirilmiştir.

### B. Tarihsel Bağlam

Sergei Mikhailovich Prokudin-Gorskii (1863-1944), Rus İmparatorluğu'nu renkli olarak belgeleyen öncü bir fotoğrafçıdır. 1909-1915 yılları arasında çektiği binlerce fotoğraf, tarihi ve kültürel açıdan büyük öneme sahiptir [1].

### C. Yaklaşım Özeti

Geliştirdiğimiz yaklaşım, brute-force arama ve metrik tabanlı optimizasyon kullanarak kanal hizalamayı otomatikleştirmektedir. NCC metrik ile piksel kaydırma değerleri hesaplanmakta, histogram eşitleme ve gama düzeltme teknikleri ile renk iyileştirme yapılmakta, ve kanallar birleştirilerek orijinal renkli görüntü elde edilmektedir.

## II. METODOLOJİ

### A. Görüntü Bölme Stratejisi

Görüntüler dikey olarak üç eşit parçaya bölünmüştür. Her parça sırasıyla mavi (B), yeşil (G) ve kırmızı (R) kanalları temsil etmektedir [2].

$$I_{height} = \frac{H}{3}, \quad I_{channel} = I[y : y + I_{height}, x] \quad (1)$$

burada  $H$  orijinal görüntünün yüksekliği,  $I_{channel}$  ise ilgili kanalı temsil etmektedir.

### B. Hizalama Algoritması Detayları

1) *Kullanılan Metrikler ve Karşılaştırması*: İki farklı metrik test edilmiş ve karşılaştırılmıştır [5]:

**SSD (Sum of Squared Differences):**

$$SSD(dx, dy) = \sum_{i,j} [I_{ref}(i, j) - I_{mov}(i + dx, j + dy)]^2 \quad (2)$$

**NCC (Normalized Cross Correlation):**

$$NCC(dx, dy) = \frac{\sum_{i,j} [I_{ref}(i, j) - \mu_{ref}][I_{mov}(i + dx, j + dy) - \mu_{mov}]}{\sigma_{ref}\sigma_{mov}} \quad (3)$$

NCC metriği parlaklık değişimlerine karşı daha robust olduğu için tercih edilmiştir.

2) *Arama Penceresi Seçimi*: Optimizasyon için  $\pm 15$  piksel arama penceresi kullanılmıştır. Brute-force arama yöntemi ile tüm olası kaydırma değerleri test edilmiştir. Kenar piksel-lerinin hatalı sonuçlara yol açmaması için %10 kenar kırpması uygulanmıştır [3].

### C. İyileştirme Teknikleri

1) *Histogram Eşitleme*: Renk kanalları arasındaki dengesizlikleri gidermek için histogram eşitleme uygulanmıştır. Bu teknik, görüntünün kontrastını artırarak daha dengeli bir renk dağılımı sağlamaktadır [4].

2) *Gama Düzeltme*: Görüntünün parlaklık seviyelerini ayarlamak için gama düzeltme ( $\gamma = 1.2$ ) uygulanmıştır. Bu teknik, görüntünün aydınlık ve karanlık bölgeleri arasındaki dengeyi iyileştirmektedir.

3) *Otomatik Kontrast Ayarlama*: Piksel değerlerini normalize ederek otomatik kontrast ayarlama uygulanmıştır. Bu yöntem, görüntünün dinamik aralığını optimize etmektedir.

## III. DENEYSEL SONUÇLAR

29 görüntü üzerinde yapılan testlerde algoritma yüksek başarı oranı göstermiştir. Örnek görüntüler için hesaplanan kaydırma değerleri ve işlem süreleri Tablo I'de sunulmuştur.

TABLE I  
ÖRNEK GÖRÜNTÜLER İÇİN HIZALAMA SONUÇLARI

Görüntü	G Kanalı (dx,dy)	R Kanalı (dx,dy)	Süre (s)
01031v	(1, 1)	(4, 2)	15.04
01167v	(2, 5)	(-2, 12)	14.87
01522v	(2, 4)	(2, 13)	15.32
00106v	(1, 3)	(3, 8)	14.95
00888v	(0, 2)	(1, 7)	15.11

Ortalama işlem süresi görüntü başına 15.2 saniye olarak ölçülmüştür. Geliştirilen pipeline, tüm görüntülerde tutarlı şekilde çalışmış ve görsel kalitede belirgin iyileşme sağlamıştır.

#### IV. TARTIŞMA VE SONUÇ

##### A. Başarılı/Başarısız Görüntüler

Algoritma, düşük seviyeli gürültüye sahip ve küçük kaymalar içeren görüntülerde yüksek başarı göstermiştir. Özellikle 01031v, 00106v ve 00888v numaralı görüntülerde mükemmel hizalama ve renk restorasyonu elde edilmiştir. Büyük geometrik distorsiyonlar içeren görüntülerde ise performans kısmen düşmüştür.

##### B. SSD vs NCC Karşılaştırması

NCC metriği, SSD'ye göre tüm test senaryolarında daha robust sonuçlar vermiştir. Özellikle parlaklık farklılıklarının olduğu görüntülerde NCC'nin normalize yapısı daha güvenilir hizalama sağlamıştır.

##### C. İyileştirme Tekniklerinin Etkisi

Histogram eşitleme ve gama düzeltme teknikleri, renk dengersizliklerinin giderilmesinde kritik rol oynamıştır. Otomatik kontrast ayarlama ise görüntülerin genel kalitesinde belirgin iyileşme sağlamıştır.

##### D. Gelecek Çalışmalar için Öneriler

- Çok çözünürlüklü piramit yaklaşımı ile hesaplama süresinin optimize edilmesi
- Derin öğrenme tabanlı otomatik renk düzeltme tekniklerinin entegrasyonu
- Daha gelişmiş geometrik dönüşüm modellerinin uygulanması
- Real-time işleme için GPU hızlandırma kullanımı
- Daha büyük tarihi görüntü arşivlerinin restorasyonu

#### KAYNAKÇA

#### REFERENCES

- [1] Library of Congress. *Prokudin-Gorskii Collection*. <https://www.loc.gov/pictures/collection/prok/>
- [2] NumPy Documentation. *NumPy Quickstart Guide*. <https://numpy.org/doc/stable/user/quickstart.html>
- [3] OpenCV. *OpenCV Python Tutorials*. [https://docs.opencv.org/4.x/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html)
- [4] Scikit-image. *Exposure and Histogram Equalization*. [https://scikit-image.org/docs/stable/auto\\_examples/](https://scikit-image.org/docs/stable/auto_examples/)
- [5] Lewis, J. P. (1995). *Fast normalized cross-correlation*. Vision Interface.
- [6] Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.
- [7] Stanford CS231n. *Computer Vision Course Materials*. <http://cs231n.stanford.edu/>
- [8] Wikipedia. *Sergey Prokudin-Gorsky*. [https://en.wikipedia.org/wiki/Sergey\\_Prokudin-Gorsky](https://en.wikipedia.org/wiki/Sergey_Prokudin-Gorsky)
- [9] Wikipedia. *Cross-correlation and Normalized Cross-Correlation*. <https://en.wikipedia.org/wiki/Cross-correlation>