第37回日本情報地球学会講演会 講演要旨集 009-010頁 2026年

# ウェブブラウザのネイティブ画像処理における RGB 値不一致とその影響

北尾　馨*

# RGB value discrepancies in web browser image processing and their implications

Kaoru KITAO*

＊ 合同会社うさぎ研究所　Laboratory of Web-based Service Technologies, LLC, 4-1, Mizukino-2, Moriya, Ibaraki 302-0121, Japan. E-mail: kitao@usaken.net

**キーワード**：色ズレ，色空間，データコンテナ，画像デコード
**Key words**：Color discrepancy, Color space, Data container, Image decoding

## １．はじめに

ウェブによる地図表現において，画像は従来，表示を目的としたものでしかなかった．しかし近年では，標高データのように画像をデータコンテナとして利用する手法が一般化している．可逆圧縮な画像形式を用いることで，RGB 値をデータとして利用することが可能となり，その同一性を前提とする設計が採用されている．

画像のデータコンテナとしての利用は標高データに限らない．例えば産業技術総合研究所が提供する 20 万分の 1 日本シームレス地質図（産業技術総合研究所，2025）では各地質単元に一意な色を割り当て，任意地点においてその色から凡例情報を取り出す手法を採用している．

しかし，近年のウェブブラウザの表示処理の影響により，特定の条件下に限り，取得された RGB 値がデータ作成時の RGB 値と一致しない事例が確認されている．この差分は視覚的には認識しにくい一方で，データとして扱う場合には値の再現性に影響を与える可能性がある．

本発表では，この色の不一致が生じる要因を整理し，利用方法によって影響の有無や重要性がどのように異なるかについて述べる．なお本件では，ウェブブラウザでの画像の取り扱いについて記述しているため，特段の指定がない限り，処理で使用する API はウェブブラウザの API，開発言語は JavaScript を使用する前提である．

## ２．問題の顕在化

本発表の契機は，画像をデータコンテナとして利用する過程において，RGB 値が元データの値と一致しない現象（以下，色ズレという）に直面したことである．色ズレ問題は，同一の処理であっても当初は特定の環境でしか観測できず，実装上の問題やデータの不整合との切り分けが困難であった．このため，原因の特定および再現条件の整理に時間を要した．

本発表は，こうした背景から色ズレ問題を特定環境固有の問題ではない現象として捉え，その影響を体系的に整理することを目的とする．

## ３．歴史的経緯

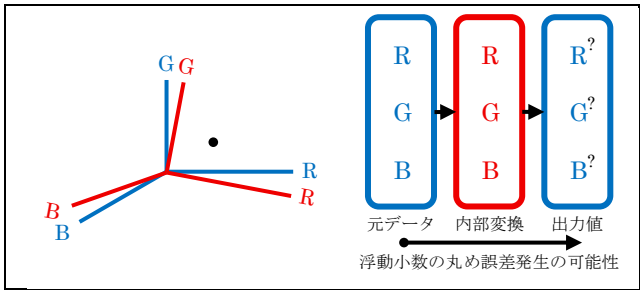
色ズレ問題は，ウェブブラウザにおける画像の内部保持処理とそのための変換等に起因する．色の取得は Canvas API 等を通じて容易に実装可能であるが，色ズレ自体は新たに発生した不具合ではなく，ウェブ技術と表示環境の変化に伴って顕在化した現象である．

2000 年代から 2010 年代前半にかけては，sRGB がウェブにおける事実上の標準色空間（色の座標系）として扱われ，ディスプレイもそれに近い特性を持つものが主流であった．このため，画像の RGB 値は実務上ほぼそのまま扱われ，Canvas 経由で取得した RGB 値も元データと一致することが一般的であり，「色＝値」として利用する前提が実務上成立していた（WebKit, 2016）．

その後，2010 年代後半になると，新たな色空間である Display-P3 等広色域ディスプレイが普及し，OS およびウェブブラウザにおけるカラーマネジメントが厳格に適用されるようになった．これにより，画像は表示環境に応じて内部色空間へ変換されることが前提となり，従来のように元の RGB 値をそのまま取得できるとは限らなくなった．さらに，内部処理で値を浮動小数点形式で扱った後，出力時に 8bit 整数へ変換する過程で丸め誤差が生じうる．

第 1 図に色空間の違いと色ズレの概念図を示す．

このように，「RGB 値はそのまま取得できる」という従来の前提は，環境の変化によって徐々に成立しなくなり，結果として現在になって色ズレ問題が明確に顕在化したと整理できる．




第 1 図　色空間（色の座標系）の違いと色ズレの概念
同じ位置（色：黒点で示す）でも座標系の違い（赤と青）で RGB の各値が異なり（左），変換で色ズレが生じる（右）．

## ４．顕在化の条件

Canvas における色ズレは，すべての環境で一様に発生する現象ではなく，複数の条件が組み合わさったときに顕在化する．主な要因は，ウェブブラウザによる色空間変換と，その後の 8bit 整数化における丸め誤差であると考えられるが，実際に問題として観測されるかどうかは実行環境に依存する．具体的には，広色域ディスプレイやカラーマネジメントが有効な環境では，画像が内部色空間へ変換される過程で浮動小数点誤差が生じ，その後の丸め処理によって最終的な整数値の差として表面化しうる．一方で，sRGB に近い条件で動作している環境では変換の影響が小さく，色ズレは顕在化しにくい．また，画像側の条件として ICC プロファイル（画像に埋め込まれた色空間情報）の有無や種類も影響する．さらに，OS やウェブブラウザの実装，GPU による描画経路の違いも微小な差分を生む要因となる．これらの条件が重なり，丸め境界を跨ぐ場合に色ズレが明確な問題として顕在化することになる．

以下，具体的な色ズレの発生過程について整理する．

### 4.1　元となる可逆圧縮画像の色データ

一般にウェブで扱う画像の色は RGB 各チャネル 8 ビットの整数値で表現され，各値はそれぞれ 0 から 255 の範囲を取る．この値は非線形なエンコード値であり，画像を Canvas に展開する際，処理に適した線形へ変換され，内部処理や描画処理で使用される．この値を再度非線形化すると理論上は元の値に戻る．

### 4.2　色ズレが発生する要因

理論上は元の値に戻るが，近似関数等の使用および精度や色空間変換等により微小誤差が生まれる．微小誤差混入の結果として変換の過程で元の値に誤差が生じ，その値が丸め境界を跨ぐか否かで誤差になるか決まる．微小誤差は値によって，以下の条件により丸め境界への近さが変わる．

暗部（元の値で 0 に近い場合）では感度が高く同じ微小誤差でも最終的な変換結果への影響が大きくなる．

明部（元の値で 255 に近い場合）では感度が小さく，同じ微小誤差でも最終的な変換結果への影響が小さくなる．

ここで「感度」とは非線形変換における誤差の増幅特性であり，非線形カーブの傾きによって誤差がどれだけ結果に影響するかを表す．変換の条件によって，値の小さい領域では傾きが大きく，同じ誤差でも結果への影響が大きくなりやすい．理論（同一条件）上，微小誤差は元の値に対して常に同じ値であるはずだが，以下の条件により微小誤差の現れ方にも差が生じる．

浮動小数演算の丸め順序の違い

内部浮動小数点精度の違い(32 ビット，64 ビットなど)

使用するガンマ変換テーブルの違い

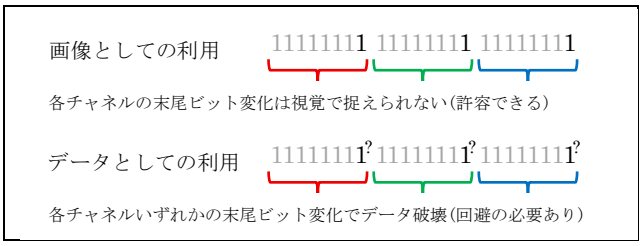
OS やブラウザによる実装差

加えて Canvas では，条件によって隣接ピクセルとの補間が行われる場合がある．これにより元の値との差(前述とは異なる処理種別に起因する広義の誤差)が生じることがあり，色ズレ発生を増加させる要因となり得るが，この点については色ズレ頻度の差の主因であるとは断定できない．

これらに起因する誤差は RGB 各チャネルで独立して現れるため，各チャネルにおいて元の値とのズレが発生しうる．

## ５．対処法

色ズレへの対処は，要求仕様に応じて段階的に整理できる．可視化用途などで誤差が許容できる場合は，従来通り Canvas を用いた処理でも問題が生じにくい．




第 2 図　用途別の対処法の違い

誤差を可能な限り抑制したい場合は，画像デコード API である createImageBitmap を介して Canvas で処理することで変換の影響を低減することが可能である．具体的なサンプルコードを以下に示す(参考：MDN Web docs, 2025)．

```
const bitmap = await createImageBitmap(blob, {
    colorSpaceConversion: "none",
    premultiplyAlpha: "none"
});
```

blob は画像データ本体，colorSpaceConversion は色空間変換の有無，premultiplyAlpha は透明度を RGB に乗算するかどうかを指定する．この結果を用いて Canvas に展開することで，より元データに近い RGB 値を取得できる．ただし，この方法でもウェブブラウザ内部の変換処理に起因する誤差を完全に排除することはできない．

RGB 値の完全一致が要求される場合には，Canvas を経由しない設計が必要となる．具体的には，JavaScript で記述された専用ライブラリや WebAssembly(ウェブブラウザ上で動作するバイナリ形式のプログラム)によって画像をデコードし，直接 RGB バイト列（実際にはこれに不透明度を加えた RGBA バイト列）を取得する方法である．これらの手法を採用することにより，Canvas 由来の変換を介さない RGBA 値を取得できる．

このように，対処法は「許容する」「低減する」「回避する」の三段階に整理でき，要求精度に応じた使い分けが重要となる(第 2 図およびサンプルコード)．

## ６．おわりに

色ズレ問題は特定環境における不具合ではなく，カラーマネジメントの普及と表示環境の多様化に伴って，従来の前提が成立しなくなったことで顕在化した現象である．重要なのは，これを一律に問題と捉えるのではなく，用途に応じて適切に扱うことである．可視化用途では従来手法や若干の工夫を加える手法が引き続き有効である一方，色をデータとして扱う用途では設計の見直しが必要となる．

今後は，画像を「表示」として扱うのか「データ」として扱うのかを明確に区別し，それに応じた処理系を選択することが重要である．

## 文　　献

MDN Web Docs（2025）Window: createImageBitmap() メソッド - Web API | MDN. https://developer.mozilla.org/ja/docs/Web/API/Window/createImageBitmap

産業技術総合研究所（2025）20 万分の 1 日本シームレス地質図. https://gbank.gsj.jp/seamless/

WebKit（2016）Introducing Wide Color on the Web. https://webkit.org/blog/6682/improving-color-on-the-web/

以上の URL は 2026 年 05 月 10 日に最終確認した．