

Poradnik

## Czy są korzyści z edycji obrazów np. 16-bitowych w GIMP-ie v2.9.3.

30-06-2016r

W tym opracowaniu nie będę wyjaśniał zagadnień związanych z terminami i teorią kolorów, w tym z głębią kolorów.

Ten temat mamy np.: <http://www.cambridgeincolour.com/tutorials/bit-depth.htm>  
<http://www.cambridgeincolour.com/tutorials/posterization.htm>  
<http://www.cambridgeincolour.com/tutorials/histograms1.htm>

Poradnik ma za zadanie pokazać zalety edycji obrazów w wyższej głębi kolorów.

### Edycja obrazów w trybie np. 16-bitowym (w przetwarzaniu końcowym)

Myślę, że udowodnię, że stosowanie do edycji Naszych obrazów, wyższej głębi bitowej niż wersji 8-bit, jest korzystne, ponieważ już konwersja do 16 bitów jest zdolna do wyświetlenia znacznie więcej kolorów, niż w wersji 8-bitowej. Ale jest również faktem, że większość zdjęć nie potrzebuje 16,8 miliona kolorów, nie mówiąc już o bilionach kolorów, aby dokładnie odtworzyć ich zawartość. Zwykle Nasze zdjęcia zawierają w najlepszym przypadku kilkaset tysięcy kolorów, choć niektóre mogą sięgać kilku milionów w zależności od motywu zdjęcia, typu aparatu - rozmiaru zdjęcia. Dodatkowo, ludzkie oko i tak nie może zobaczyć, 16,8 mln kolorów, co oznacza, że gdy umieścimy obok siebie, wersję 8-bitową i 16-bitową identycznego obrazu będą wyglądały dla nas identycznie.

Dlaczego więc, wskazanym jest lepiej pracować z obrazem np. 16-bitowym ?

Jedno słowo - **elastyczność** . Podczas edycji obrazu w programie GIMP, prędzej czy później, jeśli będziemy dokonywać kolejne drastyczne zmiany, będziemy napotkać problemy. Najczęstszym problemem jest to, co jest znane jako "*Posteryzacja*" (lub banding", gdy zgubimy tyle szczegółów w obrazie, że GIMP nie może już wyświetlać płynnych przejść z jednego koloru do drugiego. Zamiast tego pojawi się brzydkie schodkowanie, fałszywe kontury - utrata przestrzeni kolorów oraz przejść tonalnych.

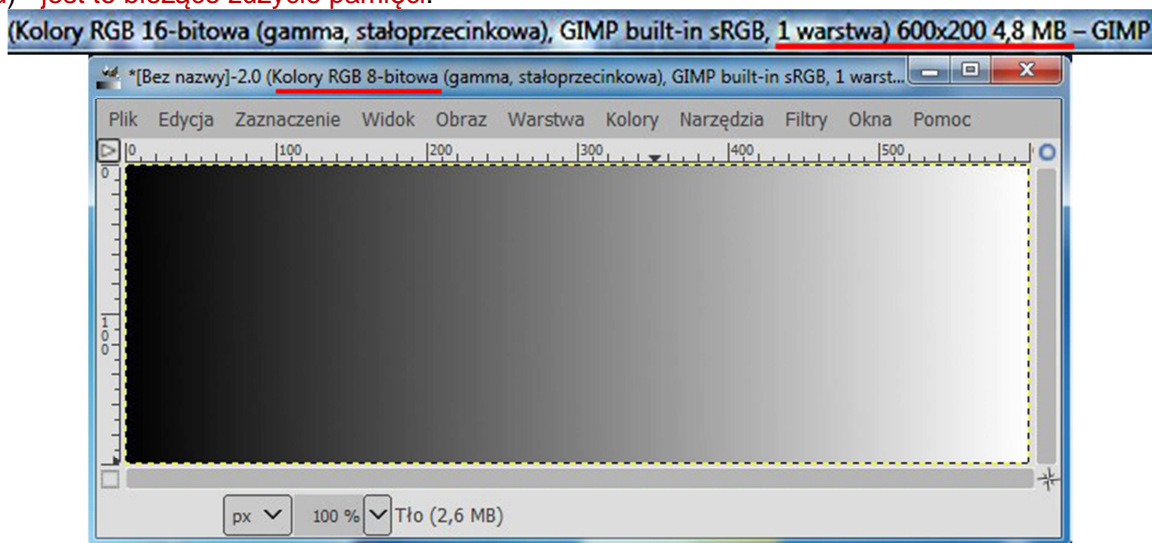
### Poniżej pokażę, co mam na myśli.

Oto dwa proste gradienty od czarni do bieli, obrazów w wymiarze 600x200 pikseli, o rozdzielczości np. 72 DPI utworzone w programie **GIMP v2.9.3**.

Oba gradienty są identyczne. Pierwszy z nich został utworzony jako obraz 8-bitowy.

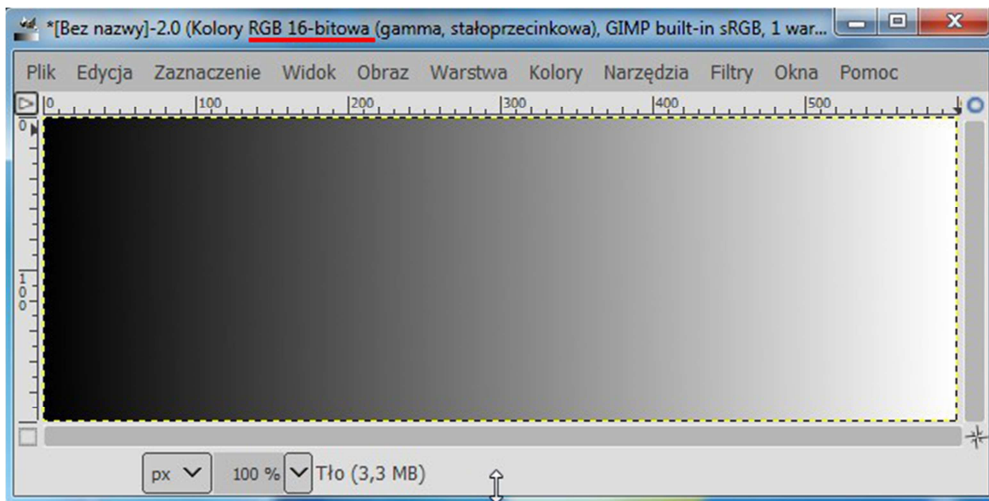
Widać liczbę "**8-bitowa**" *podkreśloną na czerwono* w górnej części okna obrazu, która mówi nam, że aktualnie warstwa obrazu jest w trybie 8-bitowym/kanal, ta warstwa zajmuje w pamięci **1,5MB**

Zobaczymy to *na pasku tytułu, (które jest wyświetlane, jeśli to ustawiono w Preferencjach, np. na pasku stanu)* - *jest to bieżące zużycie pamięci:*

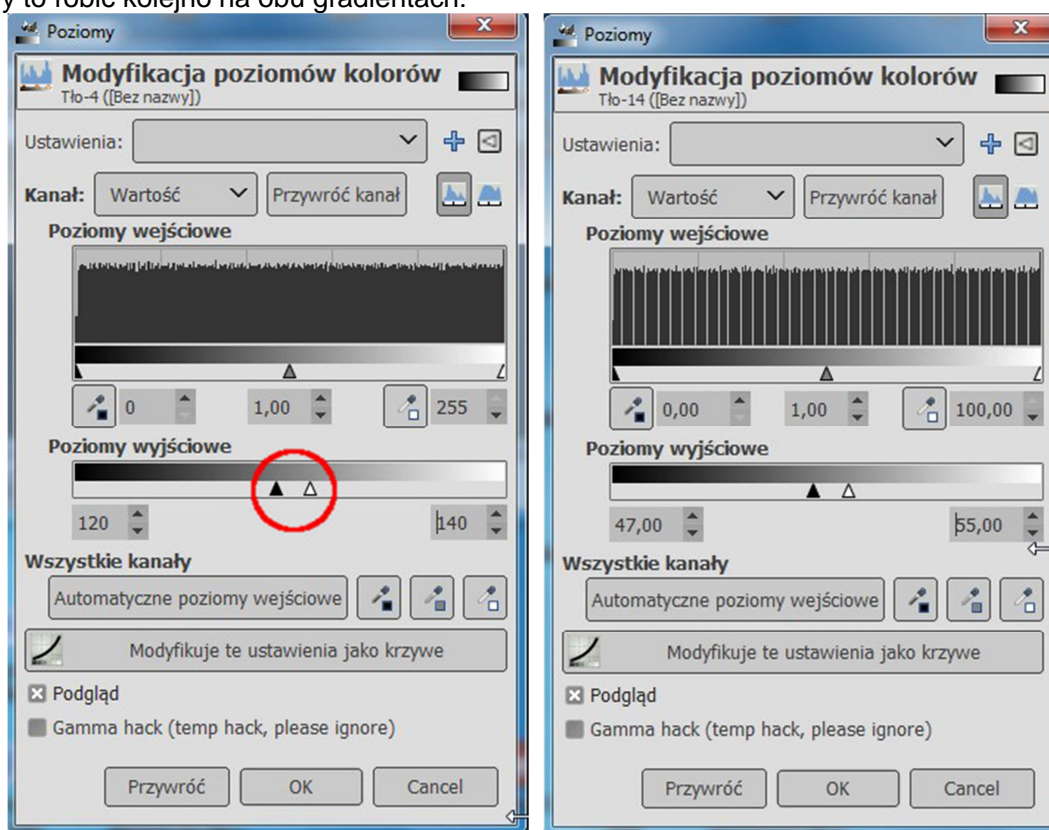


Poniżej jest dokładnie taki sam gradient utworzony jako obraz 16-bitowy.

Poza jednym faktem, że teraz widać liczbę "16-bitowa" *podkreśloną na czerwono* w górnej części okna obrazu, który mówi nam, że aktualnie obraz jest w trybie 16-bitowym, oraz że zajmuje w pamięci już **3,3 MB**, oba gradienty wyglądają tak samo:

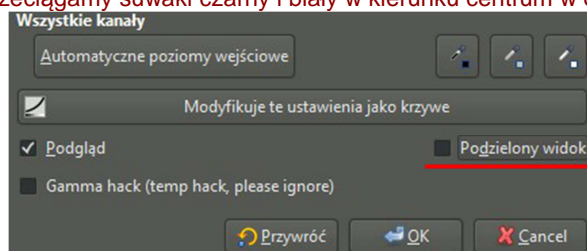


Teraz zwróćmy uwagę, co się z nimi dzieje, kiedy je edytujemy stosując **ekstremalne zmiany**. Wykonamy dokładnie te same zmiany na obu. Po pierwsze, **Regulujemy Poziomy** (Obraz => Kolory => Poziomy...) w "**Poziomy wyjściowe**" przeciągamy suwaki w kierunku centrum czarny na **120** i biały na **140**, zachowując poziomy wejściowe na **0...255**. Będziemy to robić kolejno na obu gradientach:



Okno dialogowe **Poziomy** dla 8 bit i 16 bit

Dla 16-bit skale są od 0,00 do 100:  $120/255 = 47$  i odpowiednio  $140/255 = 55$   
**"Poziomy wyjściowe"** przeciągamy suwaki czarny i biały w kierunku centrum w oknie dialogowym Poziomy.

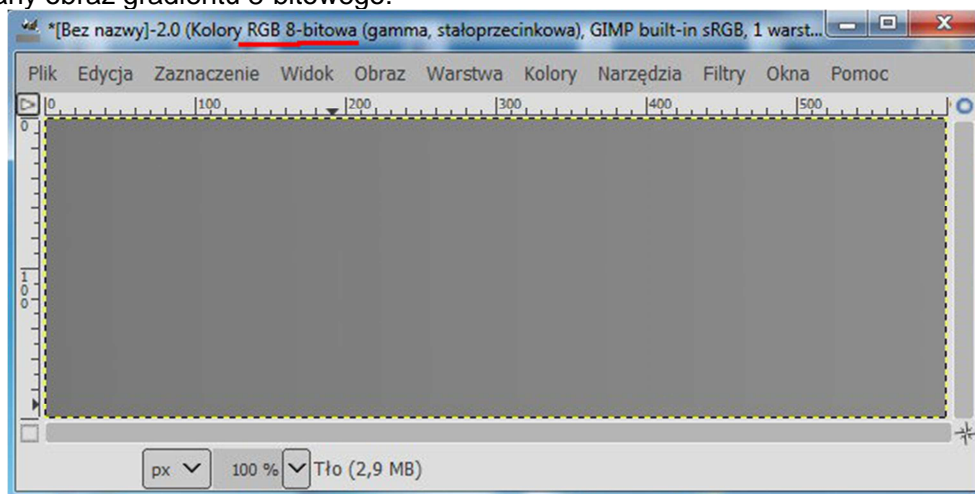


GIMP 2.9.3 Experimental portable Partha June 17, 2016.

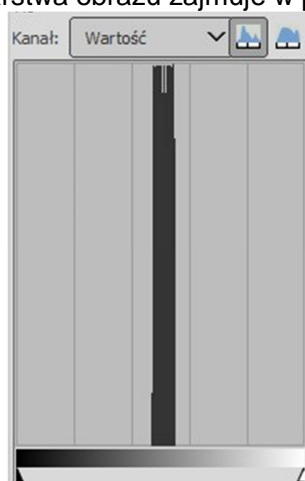
Okno dialogowe Poziomy, ma nową możliwość "Podzielony widok" w oknie obrazu.

Co tu zrobiono?. Całą gamę gradientów, od czystej czerni po lewej do czystej bieli po prawej, sflaczamy na bardzo małym odcinku w centrum, zwykle jest to obszar, w którym chcemy znaleźć średniej klasy szarość. Gradienty nie zostały faktycznie zmienione. W ten sposób umieścimy ich cały zakres tonalny w znacznie mniejszej przestrzeni. W oparciu o Histogram dokonano, modyfikacji przypisania poszczególnych wartości obrazu do danej skali odcieni szarości. Klikamy przycisk **OK**, aby wyjść z okna dialogowego **Poziomy** i teraz **spoglądamy ponownie na nasze dwa gradienty.**

Oto otrzymany obraz gradientu 8-bitowego:

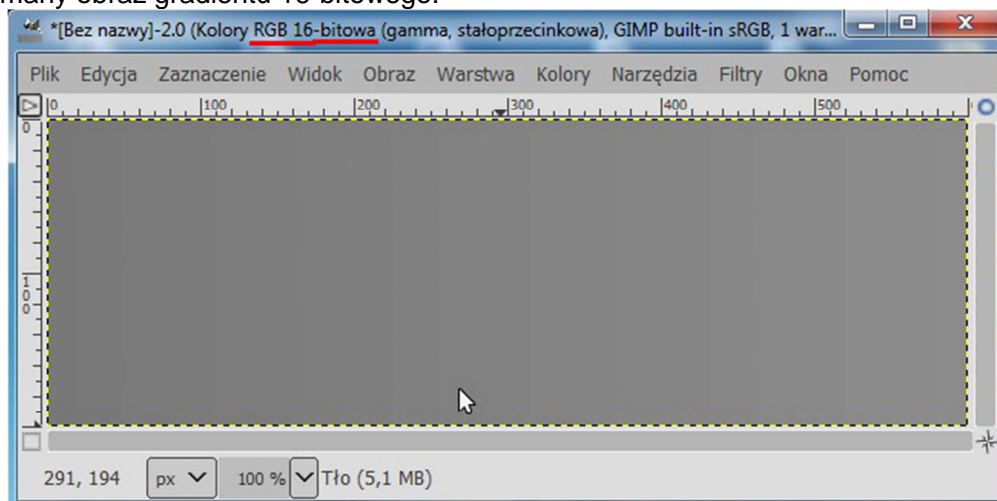


Po tej operacji warstwa obrazu zajmuje w pamięci już **1,8MB**.

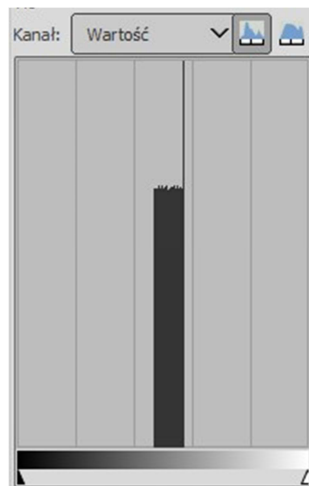


Histogram powyższego gradientu 8 bitowego (w oknie dialogowym Histogram).

A oto otrzymany obraz gradientu 16-bitowego:



Po tej operacji warstwa obrazu zajmuje w pamięci już **4,0MB**.



Histogram powyższego gradientu 16 bitowego (w oknie dialogowym Histogram).

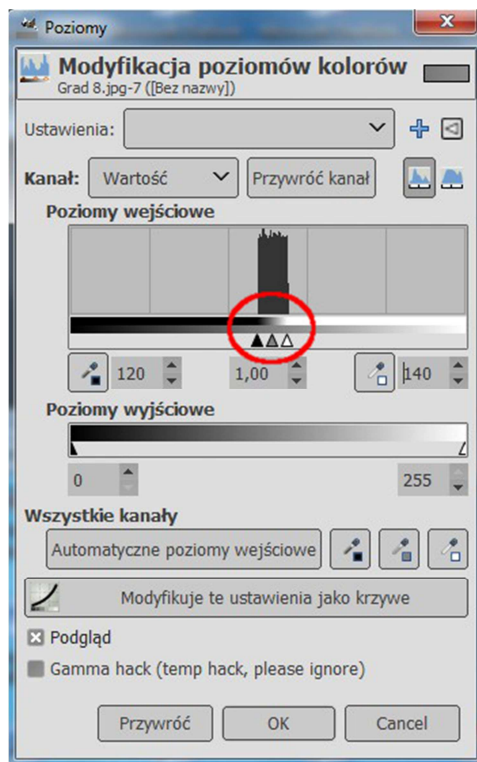
Oba gradienty po korekcie poziomów wyglądają teraz raczej jako stała szarość, ale także w tym momencie wciąż wyglądają identycznie, choć górny jest w trybie 8-bitowym, a dolny z nich jest w trybie 16-bitowym.

### Teraz spojrzmy co się dalej dzieje z naszymi dwoma gradientami,

kiedy na tym samym obrazie ponownie użyjemy poziomów rozciągając zakres tonalny gradientów **z powrotem** do czystej czerni **0**, po lewej i czystej bieli **255** po prawej stronie, ale poziomy wyjściowe pozostawiamy bez zmian.

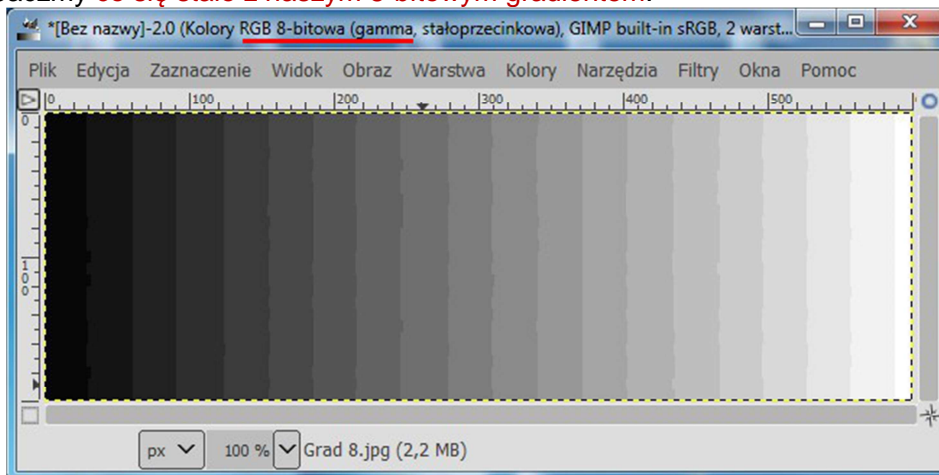
(Tworzyliśmy gradient, zastosowano Poziomy, otrzymano szarość i teraz ponownie zastosujemy do obrazu Poziomy)

W oknie dialogowym Poziomy, "**Poziomy wejściowe**" tym razem, przeciągamy suwak czarny i biały w kierunku środka, aby zmusić najciemniejsze części gradientów do powrotu do czystej czerni po lewej i najjaśniejsze części z powrotem do czystej bieli po prawej:

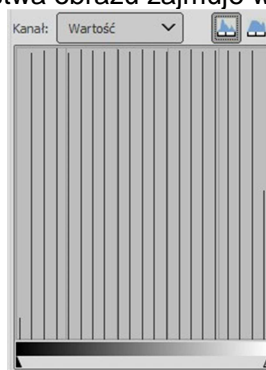


"**Poziomy wejściowe**" przeciągamy czarny i biały suwak w kierunku środka, aby rozciągnąć gradienty z powrotem do czystej czerni, po lewej i czystej bieli po prawej stronie.

Najpierw, zobaczmy **co się stało z naszym 8-bitowym gradientem**:



Po tej operacji warstwa obrazu zajmuje w pamięci już **2,2MB**.

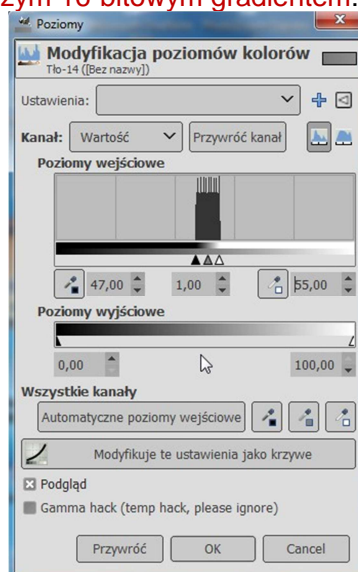


A tak wygląda Histogram ponownie otrzymanego obrazu 8-bitowego gradientu.

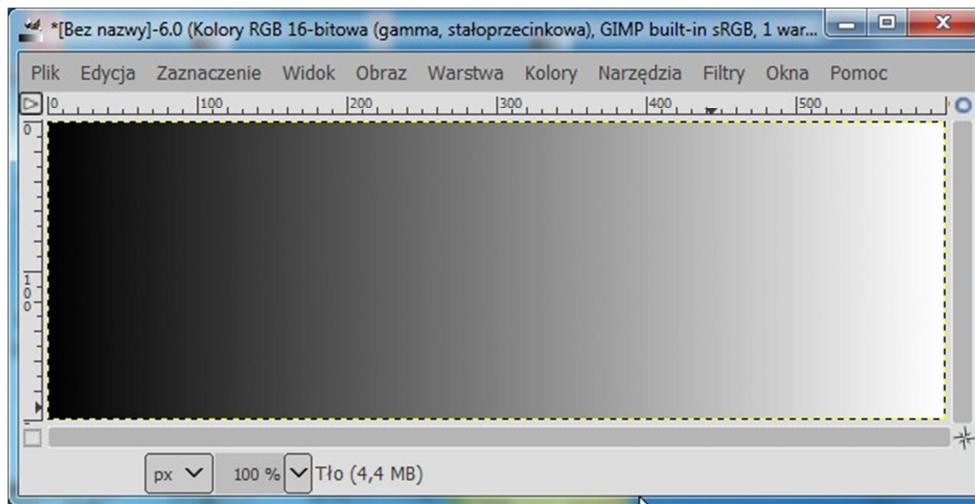
Nasz gładki gradient od czarni do bieli nie wygląda już tak gładki, a histogram poszatkowany! Gradient ma w sobie teraz **Posteryzację** (lub "banding"; "schodkowanie, fałszywe kontury - utrata przestrzeni kolorów oraz przejść tonalnych" efekt o którym wspominałem powyżej, tu można bardzo łatwo określić, gdzie występuje zmiana jednego odcienia szarości do następnego, a to dlatego, że straciliśmy ogromne kawałki szczegółów obrazu po wykonaniu tej edycji za pomocą ekstremalnych poziomów.

**W wyniku tej operacji powstaje nowy Histogram, w którym pozostaje ta sama liczba "słupków" poziomów, lecz będą one przesunięte względem pozycji słupków oryginalnego Histogramu.** Tak więc 8-bitowy obraz nie przeżył dobrze takiej ekstremalnej edycji.

A teraz zobaczmy, **co się stało z naszym 16-bitowym gradientem**:



**Przypominam:** dla 16-bit skale są od 0,00 do 100, dlatego:  $120/255 = 47$  i odpowiednio  $140/255 = 55$



Po tej operacji warstwa obrazu zajmuje w pamięci już **4,8MB**.



A tak wygląda Histogram ponownie otrzymanego obrazu 16-bitowego gradientu.

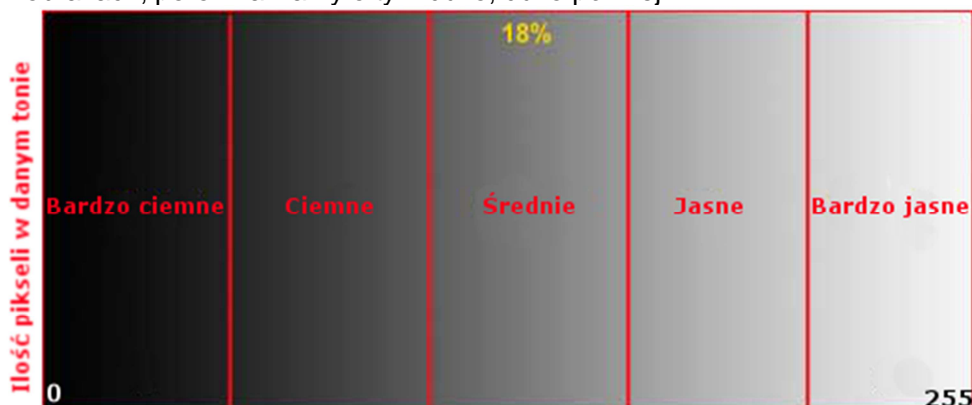
Co widzimy: nie ma luk !.

Nawet po drastycznej edycji, którą zrobiono Poziomami, gradient 16-bitowy przetrwał bez szwanku!.

### Dlaczego?

Dlaczego otrzymując ponownie gradient 8-bitowy utraciliśmy tak wiele szczegółów, podczas gdy z gradientem 16-bitowym tak się nie stało ?.

Odpowiedź kieruje nas do tego, **co już powiedziano do teraz**. 8-bitowy obraz może zawierać tylko maksymalnie 256 odcieni szarości, podczas gdy 16-bitowy obraz może zawierać maksymalnie 65,536 odcieni szarości. Choć oba gradienty wyglądały dla nas identycznie, kiedy zaczynaliśmy, te 16 tysięcy odcieni szarości dało nam dużo więcej dodatkowych możliwości i elastyczności z naszymi ekstremalnymi modyfikacjami i uczyniło znacznie mniej prawdopodobne, że nie zobaczymy później żadnych problemów w obrazie. Oczywiście, nawet w 16-bitowych obrazach, może w końcu nadejść moment, w którym stracimy dużo szczegółów, możemy zobaczyć te problemy jeśli wykonujemy mnóstwo kolejnych edycji na obrazie, ale w 8-bitowych obrazach, zjdzie to dużo szybciej, podczas gdy w 16-bitowych obrazach, porozmawiamy o tym dużo, dużo później.



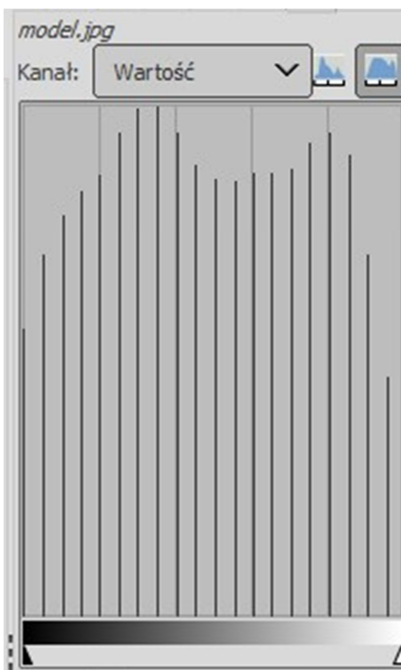
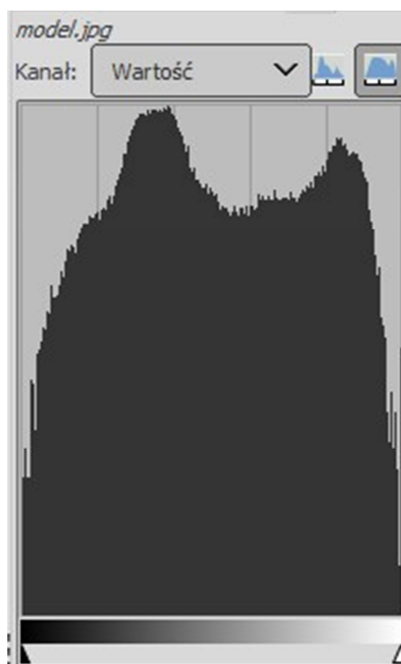
Podział okna histogramu 8 - bit

## Edycja przykładowego zdjęcia w trybie 16-bitowym

Proponuję każdemu spróbować wykonać ten sam eksperyment edycji z regulacją Poziomów na jakimś swoim kolorowym zdjęciu.

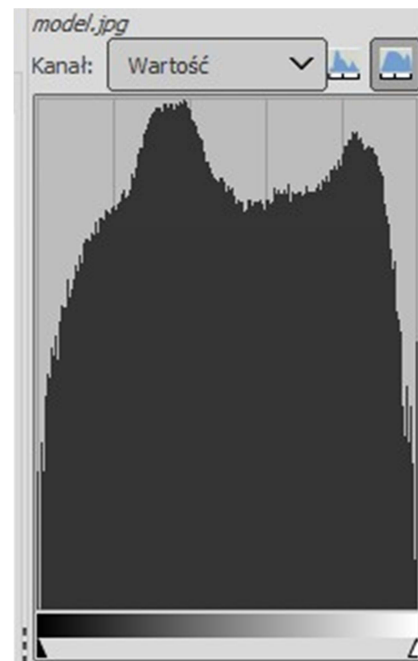
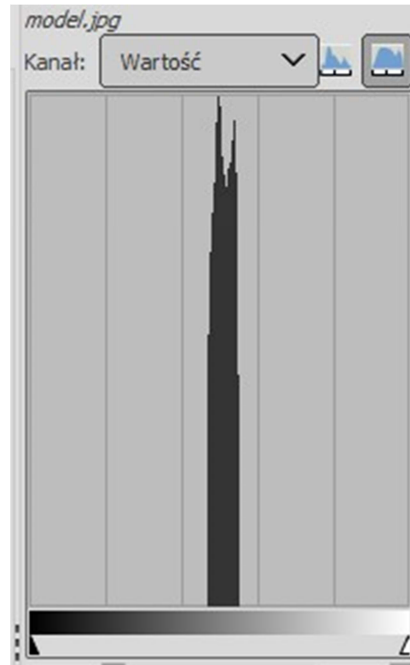
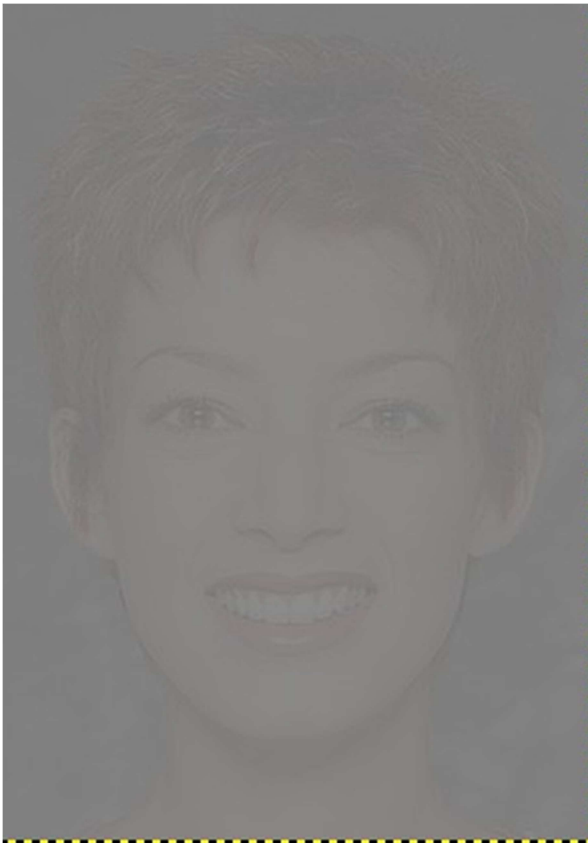
Oczywiście musimy sobie zdawać sprawę, że ta edycja, jest bardzo ekstremalna i nie może być czymś, co będziemy faktycznie stosować do swoich obrazów. Ale daje to nam jasny przykład tego, jak wiele szkód, możemy zrobić w naszych obrazach podczas edycji w wersji 8-bitowej oraz jak niewiele w porównaniu z nimi, jeśli w ogóle, wystąpi szkód gdy robimy to w wersji np. 16-bitowej.

Dla przykładu obraz 8-bit jpeg i jego Histogram przed i po jak wyżej drastycznej edycji:



Obraz 8 bit, rozciągając zakres tonalny Poziomami, tracimy informacje i Histogram staje się 'dyskretny'.

Ten sam obraz jpeg po konwersji => **Dokładność => 16-bitowa stałoprzecinkowa**  
i jego Histogram po identycznej drastycznej edycji:

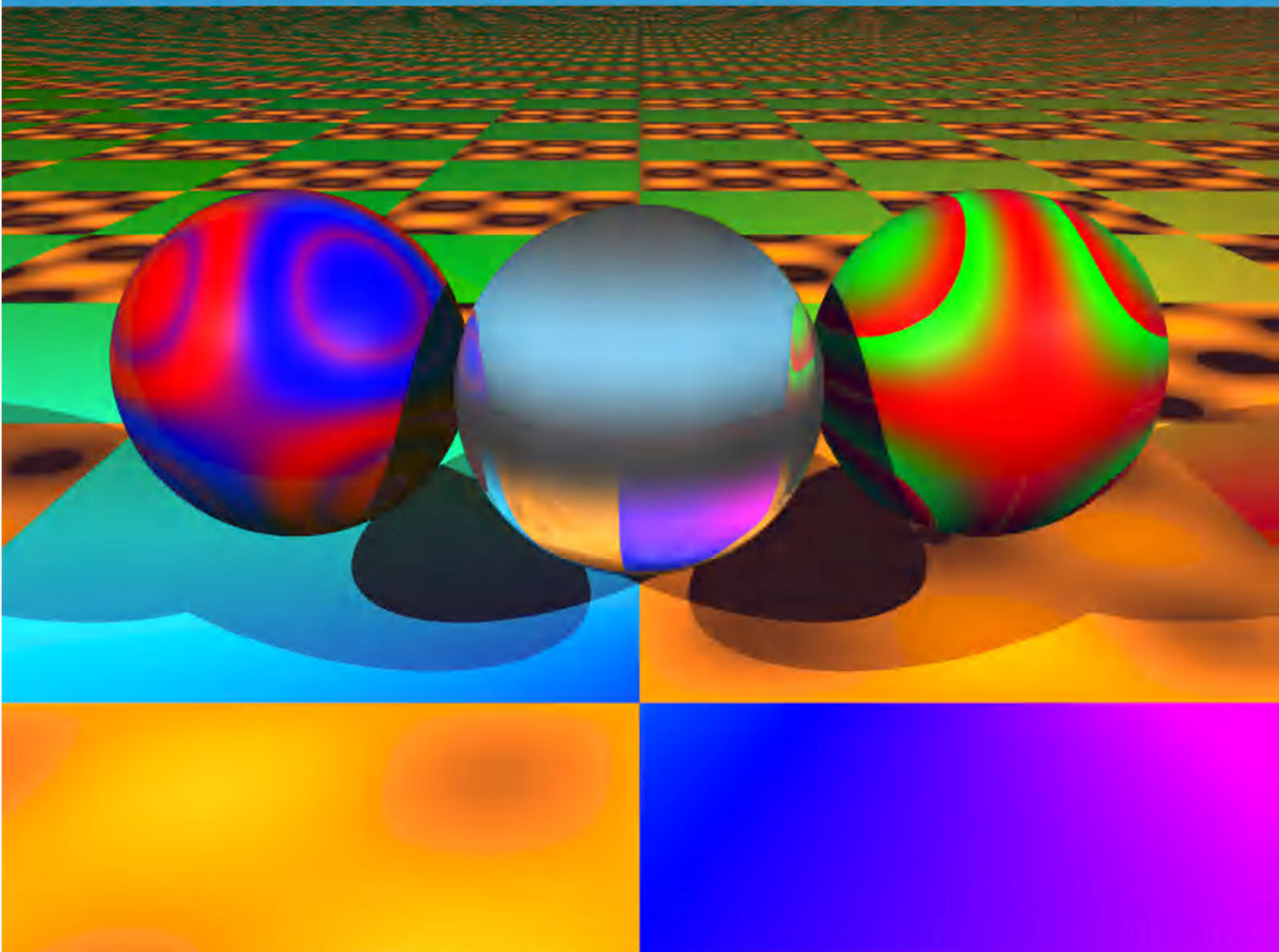


Obraz wygląda tak samo dobrze, jak przed edycją, ale jak widać z Histogramu wersja 8-bitowa straciła dużo informacji. Wszystko dlatego, że wersja 16-bitowa ma do dyspozycji ogromną ilość dostępnych

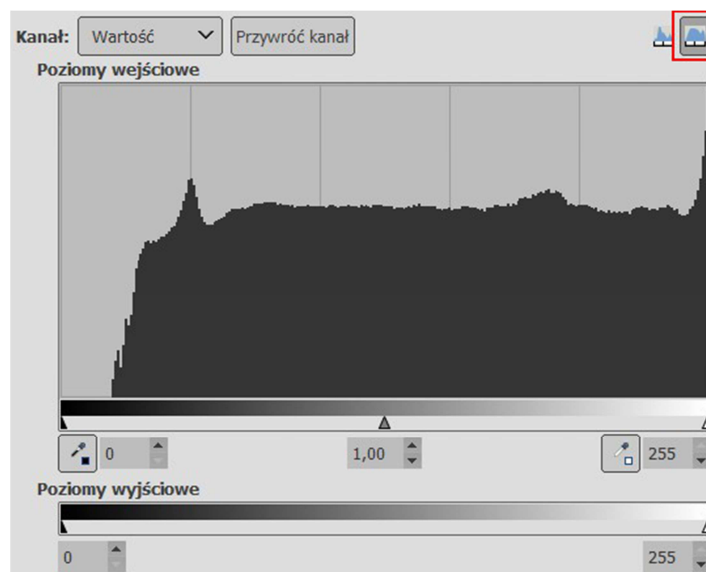


możliwych poziomów kolorów. Nawet po tak drastyczne edycji jak ta, którą wykonałem, nie byłem w stanie uczynić najmniejszego wylomu w jakości obrazu przekonwertowanego do trybu 16-bitowego. Należy pamiętać jednak, że podczas pracy z 16-bitowymi obrazami, rozmiar pliku jest dwukrotnie większy, niż z 8-bitowego obrazu, a jeśli mamy starszy komputer, może to mieć wpływ jak długo wykonuje poszczególne operacje.

**Jeszcze jeden przykład z zastosowaniem Poziomów:**

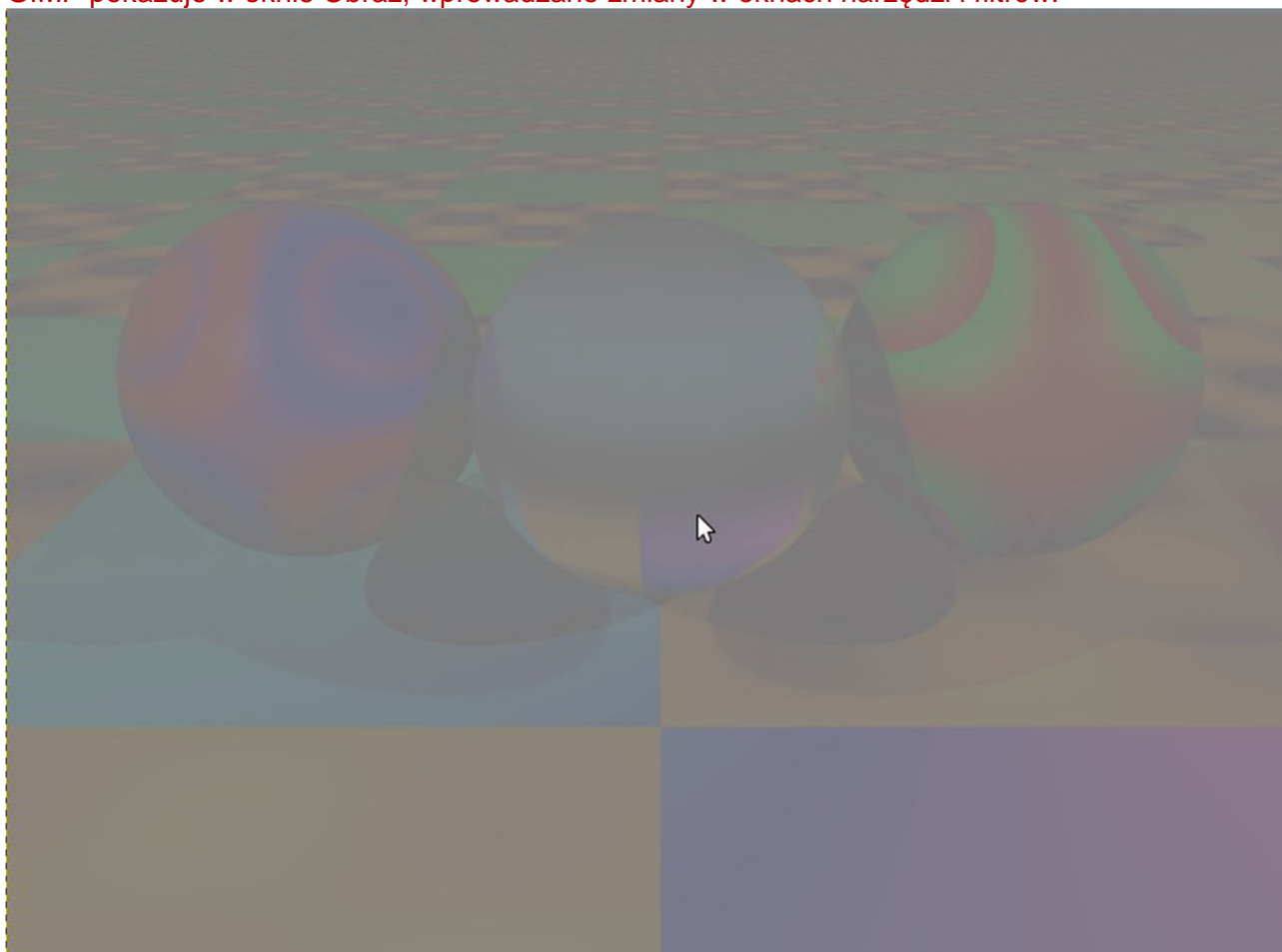


Obraz 800x600 pikseli 8-bit jpeg.

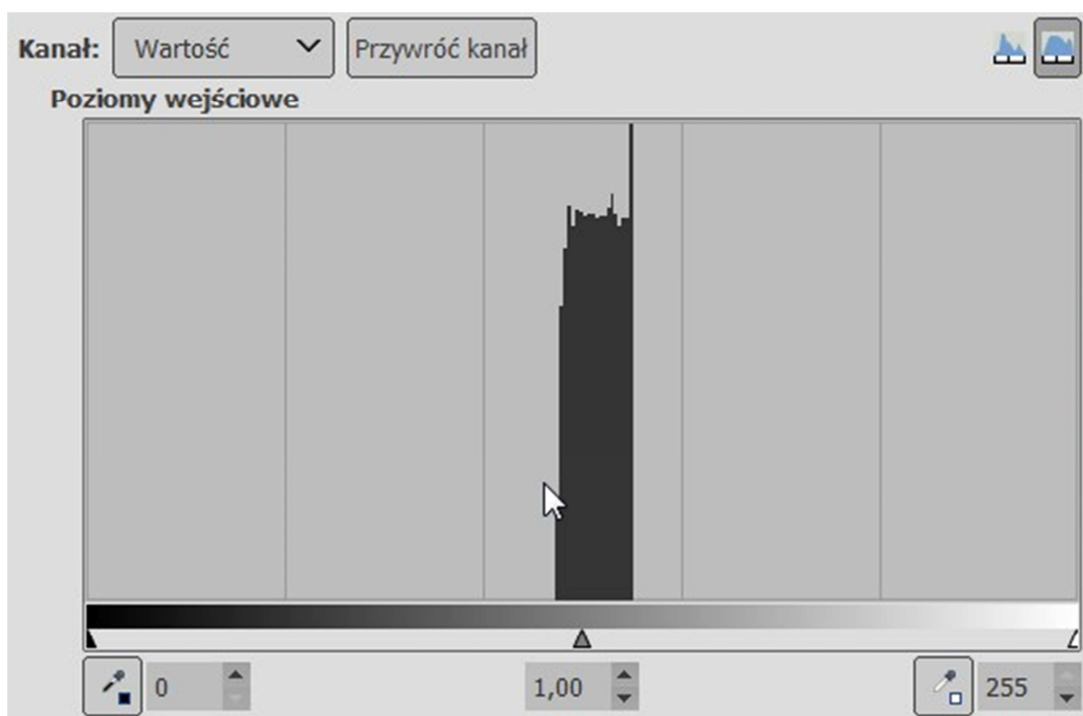


Jego Histogram oglądany w oknie dialogowym Poziomy (w skali logarytmicznej).

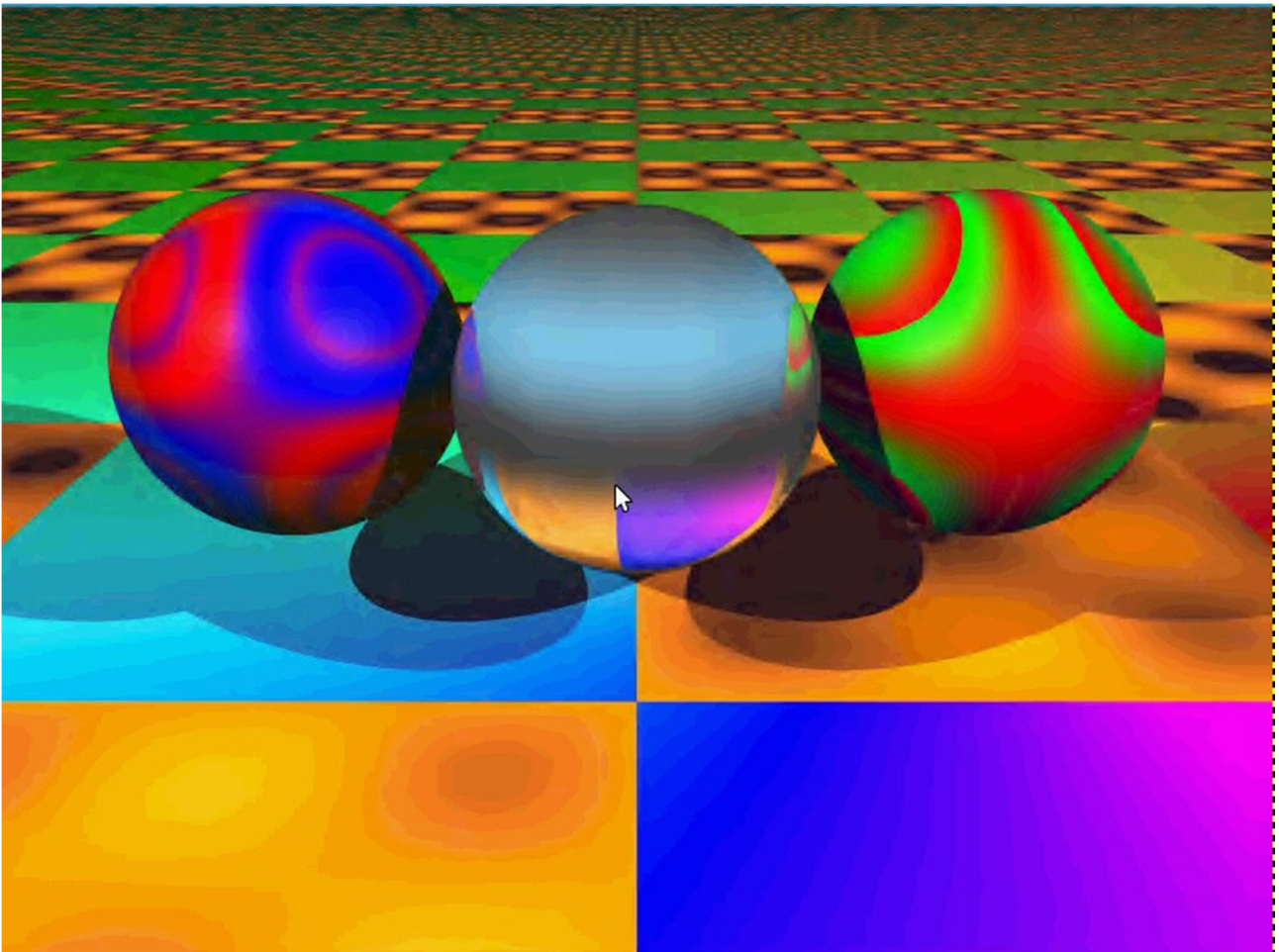
Wprowadzamy zmiany w poziomach obrazu, zatwierdzamy je i teraz **ponownie otwieramy okno dialogowe Poziomy**, by przyjrzeć się nowej postaci obrazu i histogramu. Przypominam że teraz, GIMP pokazuje w oknie Obraz, wprowadzane zmiany w oknach narzędzi i filtrów.



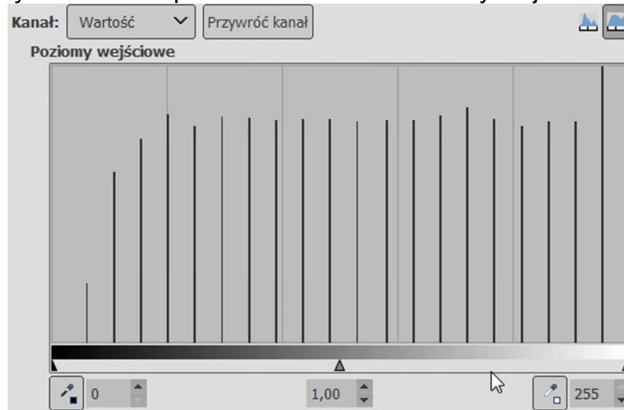
Powyższy obraz 8-bit po zastosowaniu "Poziomy wyjściowe" 120 i 140.



Histogram obrazu po zastosowaniu "Poziomy wyjściowe" 120 i 140 (po ponownym otwarciu okna dialogowego Poziomy)



Otrzymany Obraz 8-bit po zastosowaniu "Poziomy wejściowe" 120 i 140

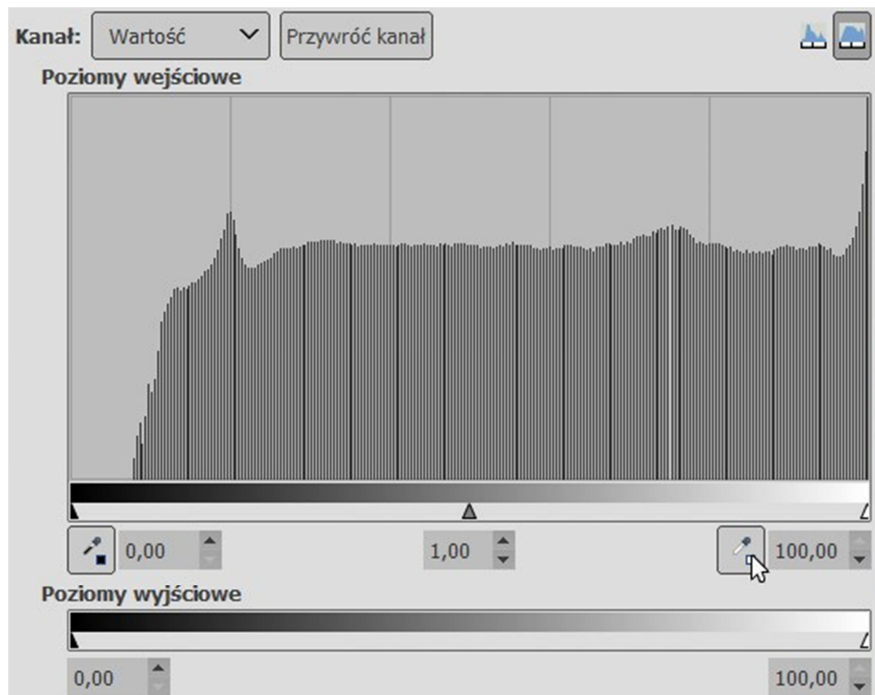


I jego otrzymany Histogram 8-bit po zastosowaniu "Poziomy wejściowe" (po ponownym otwarciu okna dialogowego Poziomy) - te szczeliny to utrata przestrzeni kolorów oraz przejść tonalnych **nie mamy możliwości, aby wymyślić nowe wartości do wypełnienia powstałych szczelin.**

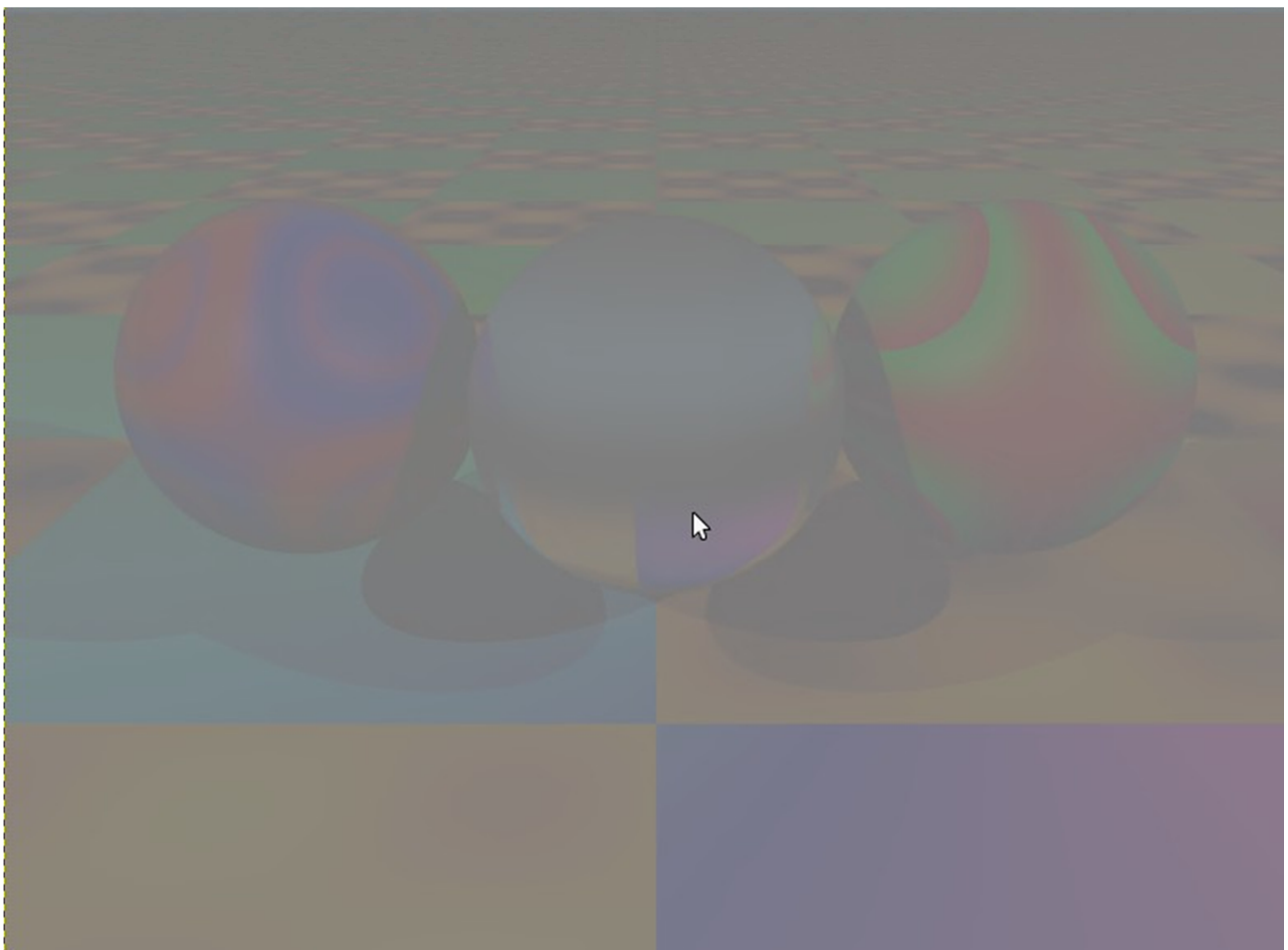


Wycinek Ponownie otrzymanego obrazu 8-bit powiększenie 200% **widoczna Posteryzacja.**

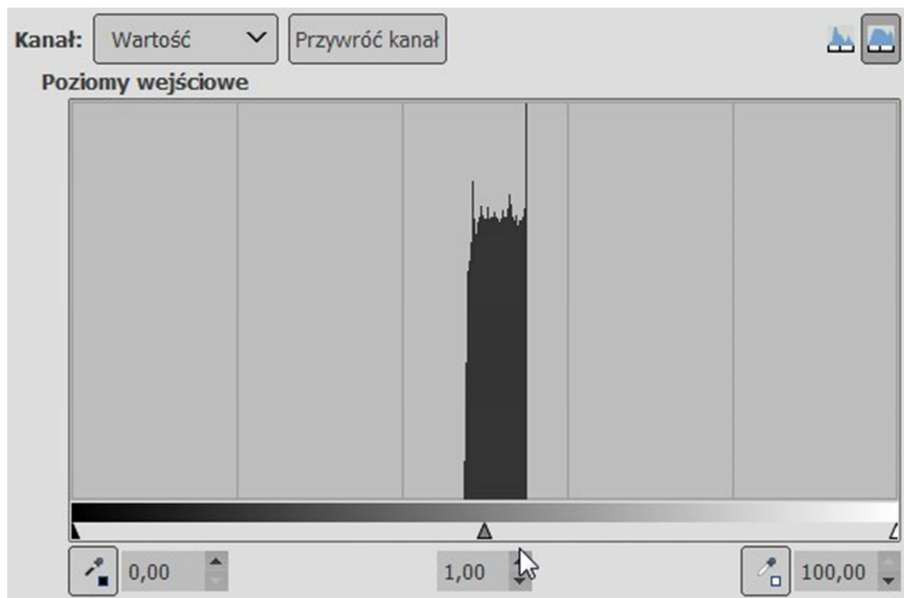
Teraz ten sam obraz, ale skonwertowany do 16-bit



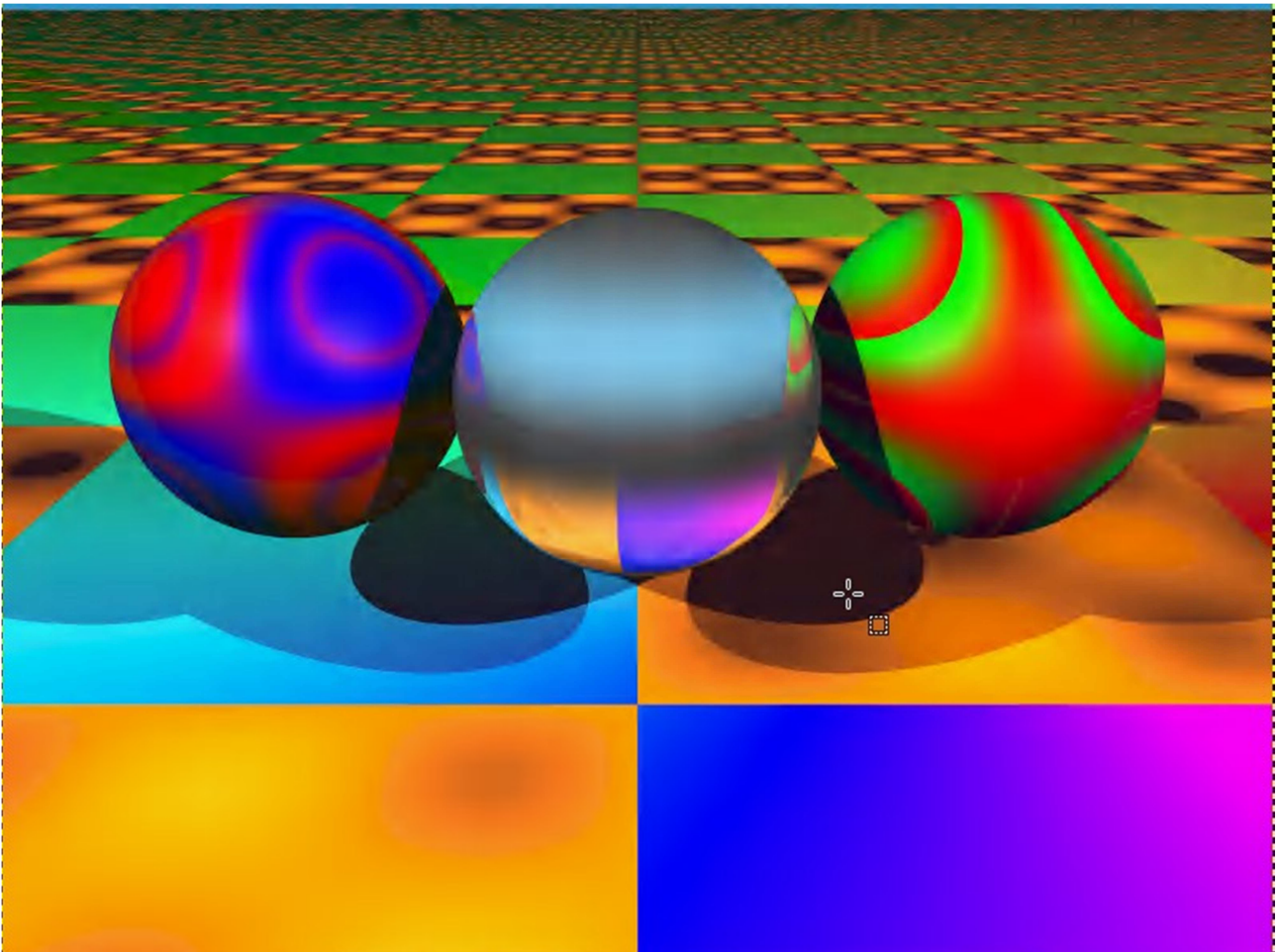
Jego Histogram oglądany w oknie dialogowym Poziomy.



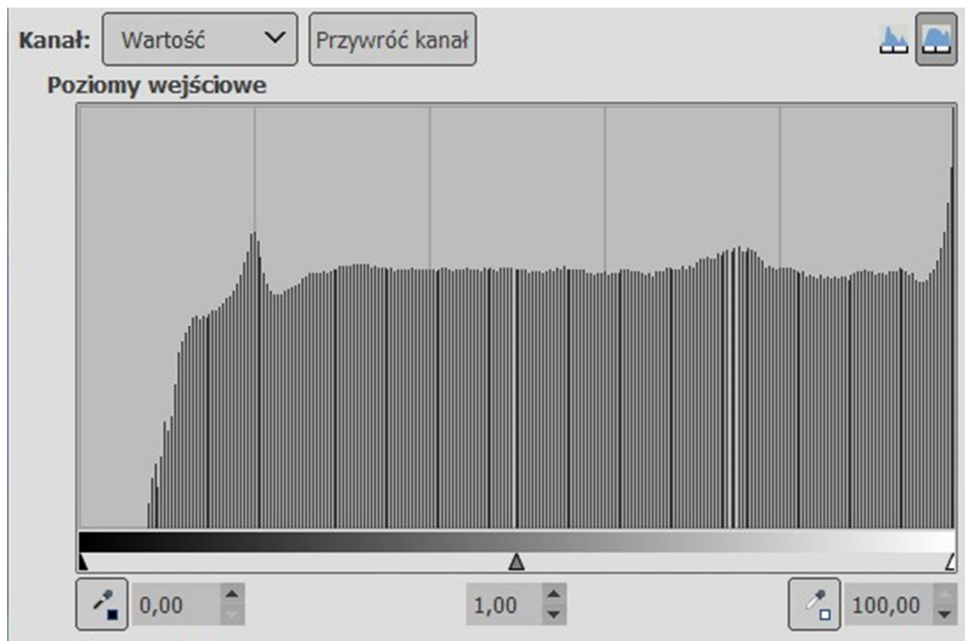
Obraz po konwersji do 16-bit i po zastosowaniu "Poziomy wyjściowe" 47 i 55.



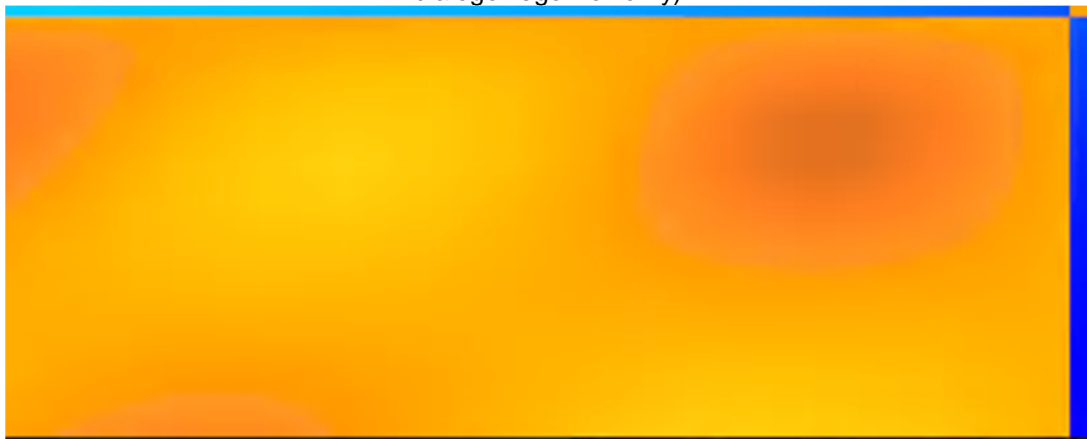
Histogram tego obrazu po zastosowaniu "Poziomy wejściowe" 47 i 55 (po ponownym otwarciu okna dialogowego Poziomy)



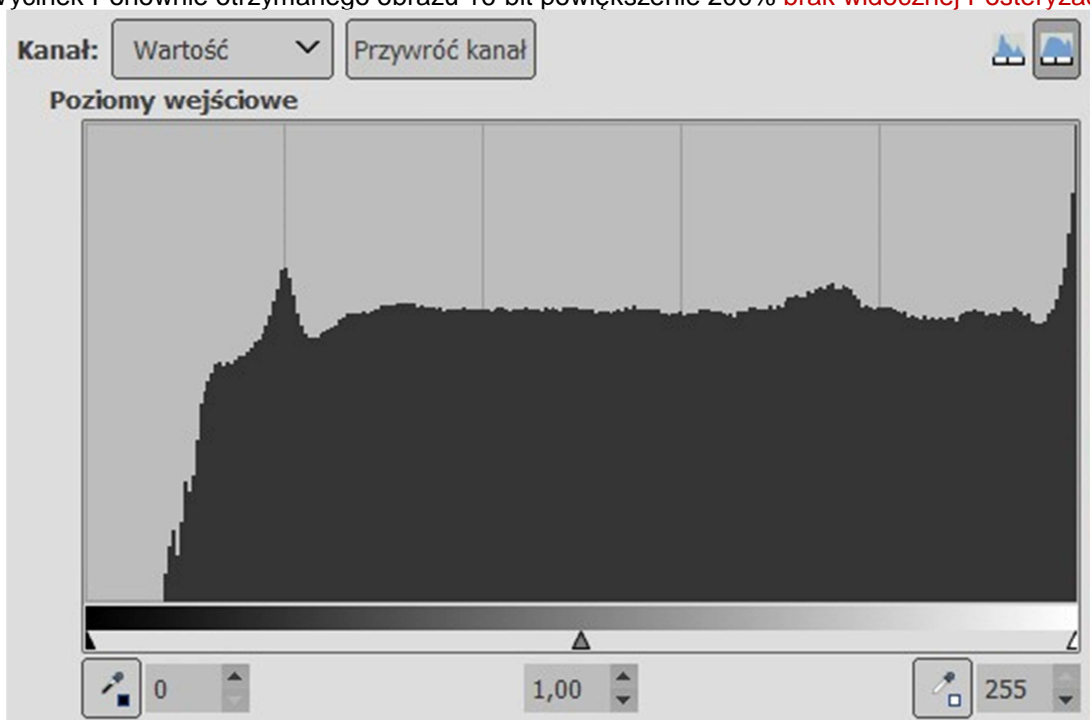
Otrzymany obraz wynikowy 16-bit po zastosowaniu "Poziomy wejściowe" 47 i 55



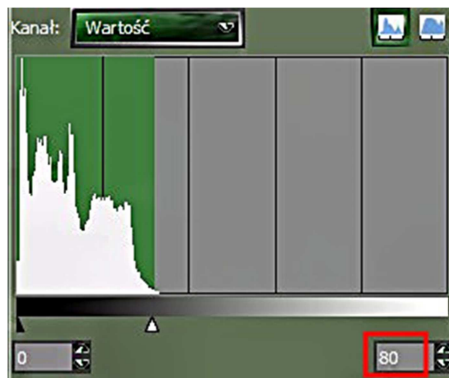
Otrzymany Histogram obrazu 16-bit po zastosowaniu "Poziomy wejściowe" (po ponownym otwarciu okna dialogowego Poziomy)



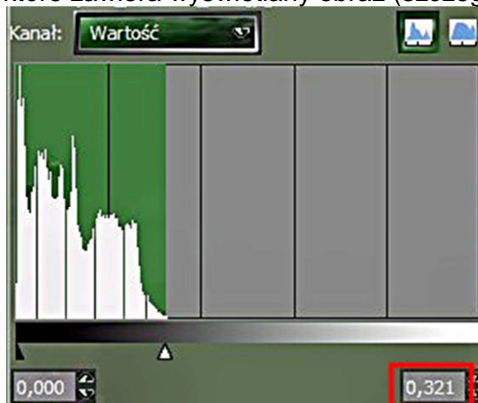
Wycinek Ponownie otrzymanego obrazu 16-bit powiększenie 200% **brak widocznej Posteryzacji.**



Histogram obrazu 16bit **po jego konwersji do 8bit** (po ponownym otwarciu okna dialogowego Poziomy)



Przesuwając w oknie dialogowym **Histogram** suwaki czarny i biały **lub** jeden z nich odczytamy wartość ilości poziomów, które zawiera wyświetlany obraz (szczegóły poniżej).



Jak już podano w GIMP-ie Histogram **np.** dla 16 bit wyświetla skalę od 0,00 do 100,00

Powyższa wersja 16-bitowa przykładowego zdjęcia miałyby  $80 \times 2^8$  poziomów, wyniesie to  $80 \times 256 = 20480$  poziomów **lub**  $80/256 = 0,3125$  **lub**  $65,536 \times 0,3125 = 20480$ .

Powyżej robiliśmy dwie kopie tego samego pliku 8-bitowego; jeden pozostanie 8-bitowym, drugi konwertujemy do 16-bit. Jeśli zrobimy **małą zmianę w obu plikach i przekonwertujemy plik 16-bit z powrotem do 8-bit. Będą wyglądać tak samo.**

Pokazane powyżej histogramy po dopasowaniu poziomów obrazu 8-bit **oraz po konwersji do 16-bit są przed zapisem!**

**Dlaczego zaznaczono przed zapisem!**

**Można zrobić szybki test:**

**Utwórzmy i zapiszmy** dwa zdjęcia 8-bitowe, przed i po **ustaleniu** histogramu poziomów. Otwieramy je w GIMP-ie. Sprawdzamy histogramy. Luki? Piki? Żadnego w obu zdjęciach... Dziwne? Niezupełnie.

Podczas interpolacji pikseli **przy zapisie**, generowane są wartości pośrednie, a to wyrównuje histogram!! Również zastosowanie nawet łagodnego odsumowania wygładzi histogram.

Możemy również spróbować jaki otrzymamy wynik po użyciu przycisku "**Automatyczne poziomy wejściowe**" poniżej histogramu w oknie Poziomy. Czasami automatyczna ekspozycja jest niemal idealna, innym razem zobaczymy niewiele lub nie ma żadnej różnicy, a dość często obraz to będzie całkowita "mgiełka". Automatyczna ekspozycja jest oczywiście bardziej skomplikowana niż mogłoby się wydawać na pierwszy rzut oka.

Lub:

[https://pl.wikipedia.org/wiki/Wyr%C3%B3wnanie\\_histogramu](https://pl.wikipedia.org/wiki/Wyr%C3%B3wnanie_histogramu) Kolory => Zrównaj

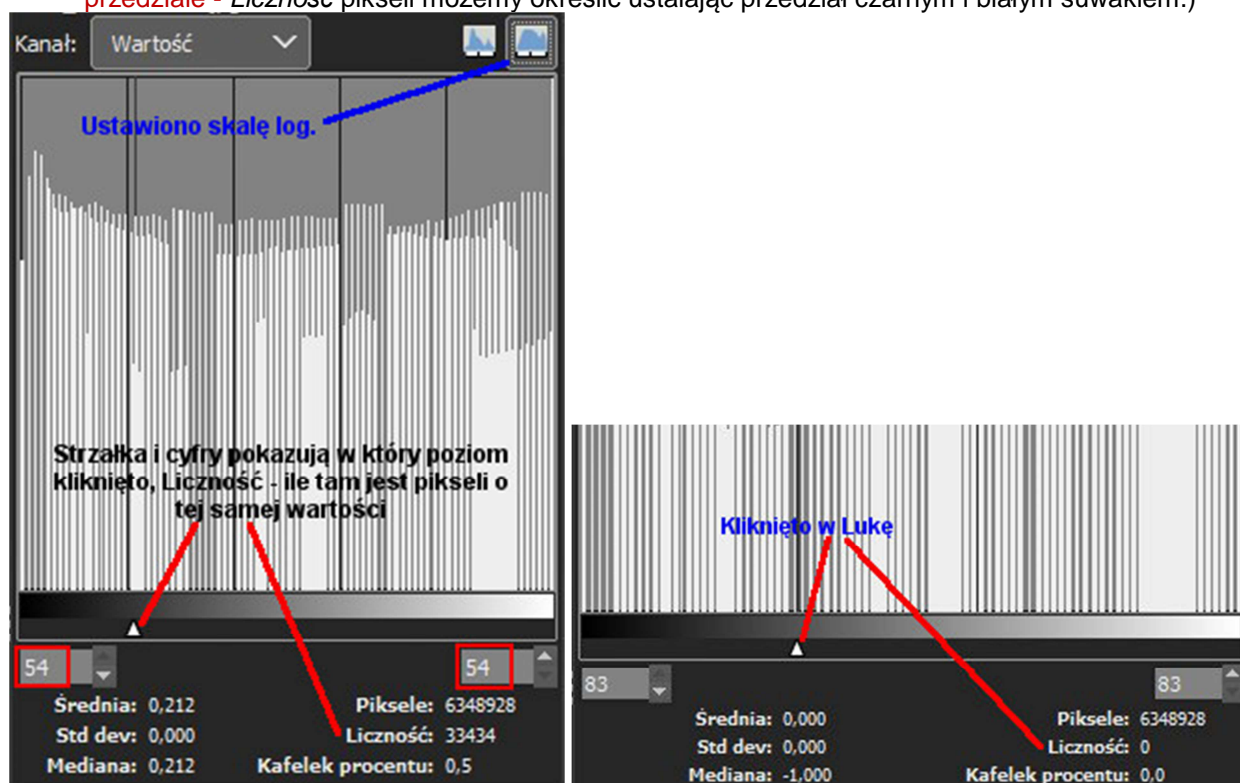
Wyrównanie histogramu obrazów kolorowych poprzez wyrównanie histogramów dla poszczególnych składowych kolorów obrazu RGB (czerwonej, zielonej i niebieskiej). Zastosowanie tej metody do kolorów składowych RGB obrazu, może powodować znaczne zmiany w balansie kolorów obrazu, ponieważ oddzielne przetwarzanie składowych spowoduje zmiany w stosunkach wartości poszczególnych kanałów. Wadą metody jest fakt, że nie rozróżnia sygnału od szumu, co może spowodować wzrost wartości szumu, przy jednoczesnym zmniejszeniu widoczności właściwego sygnału.

Wyrównanie histogramu powoduje często powstawanie nierealistycznych efektów w fotografii;

Jeśli jednak obraz zostanie najpierw przekonwertowany do innej przestrzeni kolorów, jak np. [CIE Lab](#) lub HSV, wtedy wyrównywanie histogramu może odnosić się do luminancji lub przedziału wartości, bez efektu zmiany odcienia i nasycenia obrazu.

### Uwaga:

1. dla fotografii tryb liniowy histogramu jest najbardziej użyteczny. Z drugiej strony w przypadku obrazów zawierających istotne obszary stałego koloru, to liniowy histogram często **będzie zdominowany przez pojedynczy pik** i wtedy skala logarytmiczna histogramu staje się bardziej użyteczna.
2. Nie wszyscy wiedzą, że możemy rozciągnąć okno dialogowe Histogramu (w lewo), kliknąć w pojedynczy **pik** i wtedy w dolnej części histogramu **w statystykach**, pojawi się liczba pikseli w **piku** (w **przedziale** - *Liczność* pikseli możemy określić ustalając przedział czarnym i białym suwakiem.)



Po zakończeniu edycji, widzimy luki w histogramie plików 8 bitowych. Luki oznaczają utratę poziomów w wyniku zwiększania zakresu tonalnego, natomiast skoki wartości w histogramie prezentują utratę szczegółów (niektóre piksele znajdujące się wcześniej na różnych poziomach, znajdują się teraz na tym samym poziomie i jeśli jest ich dużo na tym samym poziomie, pojawia się "pik"). Rozciągnięcie wąskiego poziom obrazu, oznacza, że między poziomami musiały zostać dodane luki na pokrycie przestrzeni tonalnej 256.

**Ale wynik, tego to w dalszym ciągu ta sama ilość poziomów.**

Jakiegokolwiek operacje korekcyjne – zmiana jasności, kontrastu, operacje na krzywych powodują „przesuwanie” danych przechowywanych z dokładnością do 8 bitów. Podczas działań arytmetycznych na danych, dokładność 8 bitów powoduje **powstawanie dużych uśrednień**, (można użyć, na przykład, 128, ale nie 128,5).

[Wartości wszystkich składowych RGB muszą być liczbami całkowitymi zarówno przed operacją, jak też po niej, czyli nie zostaną zachowane oryginalne proporcje pomiędzy składowymi RGB. Wynika z tego, że w pewnym stopniu muszą zostać zmienione kolory większości pikseli, przy czym zdecydowanie większe zmiany kolorów dotyczą obszarów ciemnych, niż obszarów jasnych (dla obszarów jasnych procent traczonej informacji o składowych koloru jest dużo mniejszy niż dla obszarów ciemnych). Przyjmijmy, że liczbę 241.5 musimy zaokrąglić do całkowitej liczby 241. Stracimy więc 0.5/241.5 zarejestrowanej wartości, czyli **stracimy tylko 2 promile składowej koloru**. Jeśli jednak ta liczba będzie miała wartość 2.5, to na zaokrągleniu stracimy 0.5/2.5, czyli **prawie 20 procent składowej koloru**.]

Jednolite przejścia tonalne są szatkowane pojawiają się "grzebień". Im więcej transformacji, tym większe dewiacje obrazu w trybie 8-bitowym.



Jeśli pracujemy w trybie 16-bitowym, zamiast tylko 256 możliwych wartości dla każdego kanału, mamy do wyboru 65,536 **czzerwieni** i tyle samo **niebieskiego** i **zielonego**.

Na pierwszy rzut oka mogłoby się wydawać, że da nam to większy zakres dynamiki. Bo zamiast zatrzymać się na 255, bylibyśmy w stanie przejść całą drogę aż do 65 tysięcy, co dawało by możliwość reprezentowania **jaśniejszej** bieli i bardziej nasyconych kolorów. **Nic z tego!**

Zamiast dodatkowego **zakresu**, dodatkowe wartości stanowią dodatkową **dokładność**.

Należy myśleć o nich jako o wartościach ułamkowych: mamy teraz **255** dodatkowych wartości **pośród** każdą z naszych wartości 8-bitowych.

GIMP 2.9.3 obsługuje wszystko, czego potrzeba w trybie Dokładności.

W nawiązaniu do **Uwagi** zamieszczonej poniżej, mogę stwierdzić, iż czas wykonana przykładowej operacji **Balansu bieli na obrazie**, w Dokładności, 32-bit stałoprzecinkowej, wynosił ok.13 sek., podczas gdy ta sama operacja wykonana w dokładności 32-bit zmiennoprzecinkowej przebiegła w **mgnieniu oka** (Obraz Histogramu otrzymamy identyczny jak dla wersji 16-bit).

Ale równocześnie mały obraz, który jako 8-bit zajmował w pamięci 24,9MB, po konwersji do 16-bit – już 57,7MB, po konwersji do 32-bit zajmował już 107,5 MB. Dlatego musimy o tym pamiętać!

Obraz 3000x2000 pikseli to kolejno ~ 123MB; ~232MB i 398MB dodanie warstwy to już 0,5GB.

Podsumowując, **rozmiar pliku** 16-bitowego obrazu jest dwa razy większy od 8-bitowego, każda dalsza rosnąca dokładność rozmiar pliku podwaja.

Ma to wpływ na szybkość przetwarzania, wymaganą pamięć RAM i pojemność naszego dysku.

### Czy więcej zawsze znaczy lepiej?

**Dzięki przekształceniu np. do 16-bit i z powrotem, nie mamy więcej tonów niż w oryginale 8-bitowym, ale istnieje więcej wartości pośrednich dostępnych dla zaawansowanej edycji. Może to pomóc w zapobieganiu degradacji obrazu w niektórych przypadkach; ponadto to zależy od tego, co się robi w edycji.**

Każda korekta tonalna wykonana w przetwarzaniu końcowym, wyrzuca informacje.

Plik 16 bit może poradzić sobie z tym dużo lepiej niż plik 8-bitowy.

Jeśli nie chcesz tracić odcieni operując na krzywych, poziomach i innych filtrach zawsze pracuj w trybie 16-bitowym.

Każdy musi sam zdecydować o tym, z jakimi **Dokładnościami** obrazów będzie pracować.

Decyzja o tym, czy pracować z obrazami 8 czy 16-bitowymi zależy od tego, w jaki sposób będziemy się nimi posługiwać, a także od tematyki, która przeważa w naszych fotografiach. Zalety obrazów 16-bitowych uwidaczniają się wtedy, gdy często pracujemy ze zdjęciami wykonanymi w trudnych warunkach oświetleniowych i robimy **znaczący** zakres edycji.

Czyli wszystko zależy od indywidualnych wymagań odnośnie końcowej jakości zdjęcia.

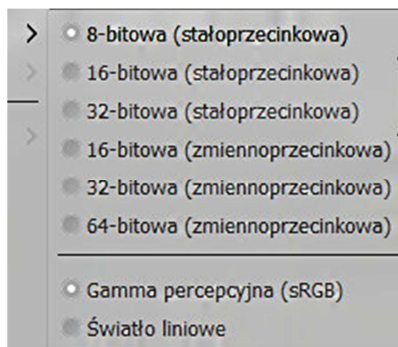
Ciekawe czy ktoś teraz jeszcze wątpi w sens konwersji np. do 16 bitów **PRZED** dokonywaniem jakichkolwiek **ekstremalnych** poprawek.

Sugeruję, aby każdy wątpiący wypróbował obydwie możliwości, po czym porównał rezultaty.

### Konwertowanie między różnymi głębiami bitowymi

❖ Wykonaj dowolną z następujących czynności w GIMP v2.9.3:

- Aby dokonać konwersji między głębią o 8 bitach na kanał a głębią o 16 bitach na kanał, wybierz polecenie Obraz => Dokładność => 16 bitów/kanał lub 8 bitów/kanał identycznie aby dokonać konwersji pomiędzy głębią o 8 bitach na kanał lub 16 bitach na kanał a głębią o 32 bitach na kanał.



Jak pokazano na powyższym zrzucie ekranu, GIMP 2.9.3 oferuje sześć różnych precyzji obrazu:

- Trzy dokładności (precyzje) *stałoprzecinkowe*: 8-bitowa stałoprzecinkowa, 16-bitowa stałoprzecinkowa, i 32-bitowa stałoprzecinkowa.
- Trzy dokładności (precyzje) *zmiennoprzecinkowe*: 16-bitowa zmiennoprzecinkowa, 32-bitowa zmiennoprzecinkowa, jak i 64-bitowa zmiennoprzecinkowa.

(Ewentualnego wyjaśnienia wymagają jeszcze przełączniki:

"Gamma percepcyjna (sRGB)" i "Światło liniowe - Linear light", można te wyjaśnienia znaleźć w:

<http://ninedegreesbelow.com/photography/users-guide-to-high-bit-depth-gimp.html> Part 2

(część 2 w/w artykułu "radiometrycznie poprawna edycja").

### Korzystanie z opcji precyzji obrazu, podczas eksportowania obrazu na dysk

Opcja menu Dokładność (precyzja) ma jeszcze jedno dodatkowe bardzo ważne zastosowanie, dyktując precyzję z jaką wyniki operacji edycji są przechowywane w pamięci RAM.

Podczas eksportowania obrazu na dysk, opcje pozwalają na precyzyjną zmianę głębi bitowej eksportowanego obrazu.

Przykładowo, niektóre edytory obrazu nie mogą odczytać zmiennoprzecinkowych TIFF.

Tak więc, jeśli chcemy wyeksportować obraz jako plik TIFF, który ma być otwarty w innym edytorze obrazów, który może odczytać tylko 8-bitowe i 16-bitowe *stałoprzecinkowe* TIFF, a Nasz stos warstw GIMP XCF aktualnie wykorzystuje 32-bitową precyzję *zmiennoprzecinkową*, możemy zmienić precyzję stosu warstwy XCF do 16-bitowej *stałoprzecinkowej* przed eksportem do formatu TIFF.

Po wyeksportowaniu obrazu, musimy pamiętać, aby kliknąć "Cofnij" ("Edycja / Cofnij..."), albo po prostu użyć skrótu klawiaturowego Ctrl-Z, aby wrócić do 32-bitowej Dokładności obliczeń *zmiennoprzecinkowych* (lub innej precyzji którą aktualnie używamy).

### Uwaga !!:

Domyślnie GIMP, **począwszy od października 2015 r** w wizualnej informacji zwrotnej histogramów Krzywych i Poziomów, a także w oknie Histogramu, **jest całkowicie mylący**, jeśli pracuje na Dokładność => "Światło liniowe".

### Określenia:

- **liczby stałoprzecinkowe** (ang. fixed-point numbers **integer**) umożliwiają zapis liczb w postaci ułamkowej, w którym pozycja przecinka ustalana jest arbitralnie w zależności od wymaganej dokładności, ta metoda jest szybsza od zmiennoprzecinkowej.
- **liczby zmiennoprzecinkowe** (ang. floating-point numbers - FP) umożliwiają obsługę większego zakresu liczb (bardzo małych lub bardzo dużych), jednak **kosztem wolniejszego** przetwarzania (chyba że nasz komputer posiada bardzo szybki koprocesor zmiennoprzecinkowy) i **jest nieznacznie bardziej dokładna** niż metoda stałoprzecinkowa. Stosowane aby ominąć błędy zaokrąglenia. Termin liczby „zmiennoprzecinkowe” oznacza, że nie istnieje stała liczba cyfr przed przecinkiem i po nim. Zapis zmiennoprzecinkowy wymaga znacznie mniejszej liczby znaków niż stałoprzecinkowy przy zapisie bardzo dużych lub bardzo małych liczb.

### Literatura uzupełniająca:

OneDrive: <http://1drv.ms/1O74Jqj> lub

[http://zbyma.gimpuj.info/Poradnik\\_Nowe\\_mo%C5%BCliwo%C5%9Bci\\_GIMP%20v.2.9.3.pdf](http://zbyma.gimpuj.info/Poradnik_Nowe_mo%C5%BCliwo%C5%9Bci_GIMP%20v.2.9.3.pdf)

### Uwaga:

Ponownie zwracam uwagę na ważne stwierdzenia pani **Elle Stone** zawarte w:

#### User's Guide to High Bit Depth GIMP 2.9.2 Part 1

<http://ninedegreesbelow.com/photography/users-guide-to-high-bit-depth-gimp.html>

### Którą Dokładność należy wybrać do edycji?

Jeśli masz szybki komputer z dużą ilością pamięci RAM, to polecam zawsze przed rozpoczęciem edycji przekształcić swoje zdjęcia na 32-bitowe zmiennoprzecinkowe.

Oto dlaczego:

1. **Niezależnie od precyzji jaką wybierzesz, wszystkie przetwarzania wewnętrzne babl/GEGL/GIMP odbywają się w 32-bitowych zmiennoprzecinkowych.** *Przeczytaj to zdanie trzy razy.*
2. **Oraz [small speed penalty for not using 32-bit floating point precision](#) >**

[ GIMP wykorzystuje wewnętrznie przetwarzanie (linear floating-point) liniowe 32-bitowe zmiennoprzecinkowe i praca na obrazie z tą precyzją jest radykalnie szybsza, niż w jakiegokolwiek innej dokładności. Na drugi dzień mierzenia stwierdziłem że dodanie warstwy alfa do obrazu z 16-bitową precyzją trwało około 22 sekund (!!!), robiąc to samo na obrazie, po pierwszym przekształceniu go do przestrzeni 32-bitowej zmiennoprzecinkowej trwało 1 sekundę! ]

**3. Opcja menu Dokładność dyktuje ile pamięci jest używane, do przechowywania w pamięci RAM, wyników obliczeń wewnętrznych:**

- o Wybór 32-bitowej precyzji zmiennoprzecinkowej pozwala na pełne wykorzystanie 32-bitowego przetwarzania zmiennopozycyjnego GEGL.
- o Jeśli pracujemy na komputerze, z niewielkim RAM-em, wydajniej będzie skorzystać z systemu 16-bitowego zmiennoprzecinkowego lub dokładności stałoprzecinkowej, ale oczywiście ceną jest utrata dokładności, gdy w nowych operacjach edycji, wykorzystywać wyniki poprzednich edycji, przechowywanych w pamięci.
- o Dla bardzo małych układów pamięci RAM, wydajność skorzysta jeszcze z wykorzystania 8-bitowej precyzji stałoprzecinkowej. Ale jeśli używamy 8-bitowej precyzji stałoprzecinkowej, usuwamy większość zalet pracy z edytorem obrazu o wysokiej głębi bitowej.
- o 64-bitowa dokładność jest dostępna głównie w celu dostosowania importu i eksportu bardzo precyzyjnych obrazów o wysokiej dokładności bitowej **do edycji naukowej**. *Nie uzyskamy żadnej obliczeniowej dokładności z wykorzystaniem 64-bitowej dokładności dla rzeczywistej edycji*. Jeśli wybierzemy 64-bitową dokładność edycji, wszystko co naprawdę robi to marnowanie zasobów systemu RAM.

**Opracowanie:**  
**Małach Zbigniew**  
**Zbyma72age**

*Wszelkie prawa zastrzeżone.*

***Poradnik nie może być publikowany w całości lub fragmentach na innych stronach www lub prasie, bez***

***wcześniejszego kontaktu z autorem poradnika i pisemnej zgody na publikację***