

Matematyk w cyfrowej ciemni fotograficznej

Marcin KYSIAK, Warszawa

Jest to tekst nawiązujący do odczytu autora na XLIV Szkole Matematyki Poglądowej *Do czego to się przydaje?*, Sulejów, styczeń 2010. Ze zrozumiałych względów (np. brak koloru) całość poruszanej problematyki nie mogła zostać przedstawiona.

Od czasu gdy fotografię zdominowały aparaty cyfrowe, operacje wykonywane przy wywoływaniu materiału i robieniu powiększeń zastąpiły korekty zdjęć dokonywane w programach graficznych. Sama cyfrowa natura obrazu poddawanego modyfikacji sugeruje, a wręcz i wymusza obecność matematyki w tych procesach. Najbardziej zaawansowana matematyka ukryta jest oczywiście w stosowanych algorytmach, na przykład metodach kompresji; my przyjrzymy się jednak takiej matematyce, której użytkownik „cyfrowej ciemni” może niemal dotknąć. Nie będzie to matematyka w sensie twierdzeń i ich dowodów, ale w sensie precyzyjnego języka służącego do opisu zjawisk czy wykonywanych modyfikacji.

Dla uproszczenia mówić będziemy wyłącznie o zdjęciach czarno-białych, a dokładniej – zdjęciach w odcieniach szarości. Zdjęcie takie mamy na ogół zapisane w postaci pliku, który po otwarciu ma kształt prostokąta złożonego z pikseli (na przykład 3456 na 2304 piksele), każdemu z pikseli przypisana jest liczba całkowita z zakresu $0, \dots, 255$ określająca jego jasność. Każdy z pikseli może mieć zatem jedną z $2^8 = 256$ możliwych jasności, gdzie wartość 0 oznacza piksel czarny, 255 – biały, a wartości pośrednie kodują różne odcienie szarości.

Oglądamy nasze zdjęcie i stwierdzamy, że jest zbyt ciemne i zbyt mało kontrastowe. Wyszukujemy więc w programie graficznym opcje JASNOŚĆ I KONTRAST i przesuwamy te suwaki w prawo. Oglądamy efekt i wtedy na ogół następuje rozczarowanie – jeżeli zadziałaliśmy nieostrożnie tymi suwakami, efekt nie jest specjalnie ciekawy: na zdjęciu mogą pojawić się białe albo czarne placki w obszarach, gdzie jeszcze przed chwilą było coś widać. Dlaczego?

Korekta jasności i kontrastu suwakami bywa operacją bardzo destruktywną, a w dodatku nad dokonywaną destrukcją mamy bardzo ograniczoną kontrolę. Zastanówmy się, na czym może polegać zwiększenie jasności pikseli następujące po ustawieniu suwaka JASNOŚĆ np. na wartość +30. Otóż programy graficzne zachowują się w takich sytuacjach dwojako. Gorsza opcja jest taka, że wartości wszystkich pikseli zostaną zwiększone o 30, zgodnie ze wzorem $f(n) = n + 30$. Nie dość, że w naszym obrazie po tej operacji na pewno nie będzie pikseli o wartościach mniejszych od 30 (bo minimalna wartość wyjściowa piksela to przecież 0), to jeszcze piksele o jasnościach 225 i większych otrzymają jedną i tę samą wartość 255 (bo większej przecież nie mogą!). W efekcie tam, gdzie na zdjęciu widniały na przykład jasne cumulusy albo włosy ładnej blondynki pojawi się jednolita biała plama, a to nie wygląda dobrze. Inne programy zachowują się bardziej przyzwoicie, mianowicie przy zwiększeniu jasności o N , jasność piksela wyliczana jest zgodnie ze wzorem

$$f(n) = \frac{255 - N}{255}n + N,$$

oczywiście po zaokrągleniu wyniku do wartości całkowitych. Innymi słowy, piksele o wartości 0 odwzorowywane są na piksele o wartości N , piksele o wartości 255 pozostają bez zmian, a cała reszta odwzorowywana jest funkcją liniową wyznaczoną przez te dwie wartości. Nieuniknione jest jednak, dla $N > 0$, że w obrazie nie ma po tej operacji czystej czerni, a jedynie szarość. Analogiczne problemy mogą wystąpić przy przyciemnianiu zdjęcia – albo wiele względnie ciemnych pikseli zostanie odwzorowanych na czystą czerń i w ciemnych obszarach znikną widoczne uprzednio szczegóły, albo przynajmniej ze zdjęcia znikną nam biele i zostaną same szarości.

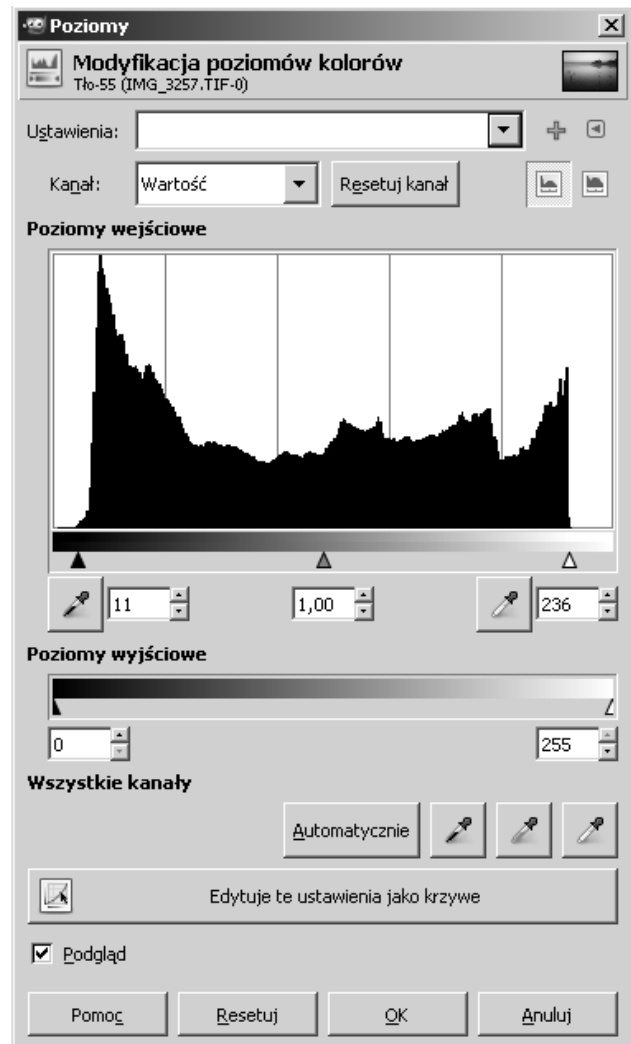
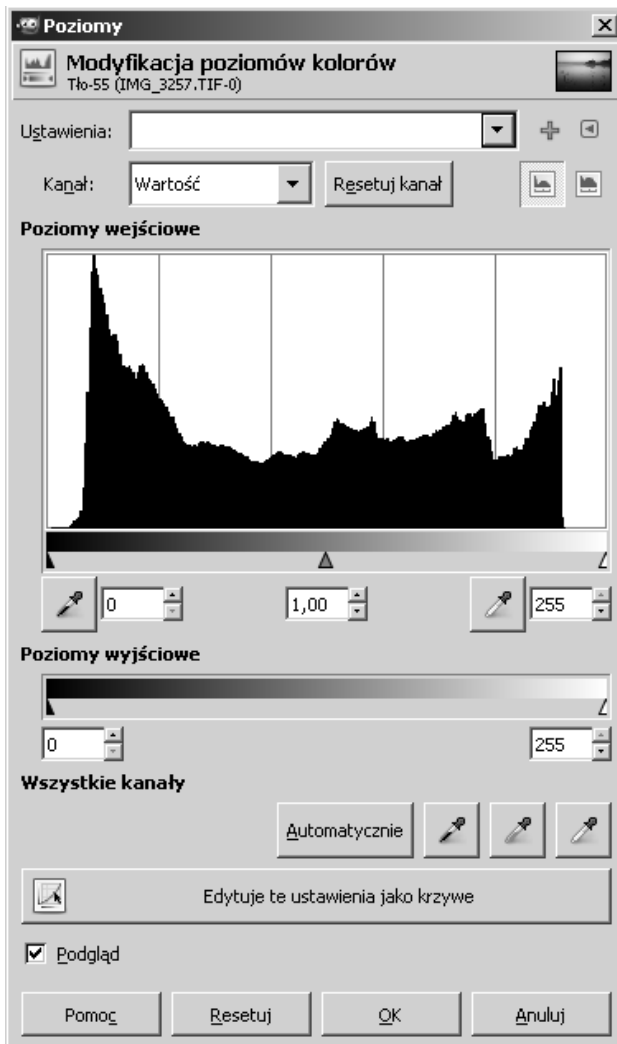
Jeszcze łatwiej zniszczyć obraz suwakiem kontrastu. Intuicyjnie zwiększenie kontrastu polega na rozjaśnieniu pikseli jasnych i przyciemnieniu ciemnych. Łatwo jednak o sytuację, gdy poprzez takie rozjaśnianie czy przyciemnianie, mówiąc potocznie, „skończy się skala” i w obszarach, gdzie było wcześniej zróżnicowanie tonalne, uzyskamy jednolitą biel lub czerń.

Zacznijmy od właściwego wyregulowania kontrastu. Z pomocą przyjdzie nam każdy odpowiednio zaawansowany program do obróbki zdjęć, na przykład darmowy GIMP dostępny na stronie <http://www.gimp.org>. Otwórzmy w nim nasze zdjęcie i wybierzmy opcję KOLORY → POZIOMY. Ujrzymy okno dialogowe, a w nim najważniejszą informację o obrazie – histogram. Ponad widoczną skalą szarości od 0 po lewej do 255 po prawej widzimy wykres, który informuje nas, jaka liczba $H(i)$ pikseli o danej jasności i występuje w obrazie. Przyjmijmy oznaczenia

$$m = \min \{i : H(i) > 0\}, \quad M = \min \{i : H(i) > 0\};$$

innymi słowy m to jasność najciemniejszego piksela w naszym obrazie, a M – najjaśniejszego. Jeżeli nasze zdjęcie cierpi na niedostatek kontrastu, jest duża szansa, że nie ma w nim pikseli o wartościach zbliżonych do 0 ani 255, czyli $m \gg 0$ i $M \ll 255$. Obrazowo – histogram jest wąski, nie wypełnia całej dostępnej mu szerokości, ale zaczyna się daleko od lewej krawędzi okna i kończy wyraźnie przed prawą. Przesuńmy teraz biały i czarny trójkącik pod skalą poziomów wejściowych w miejsca, gdzie zaczyna się i kończy histogram – w naszych oznaczeniach odpowiednio do punktów m i M (zob. rys. 1). Zdjęcie powinno nabrać wyraźnego kontrastu.

Przez znak podwójnej nierówności rozumiemy „znacznie mniejsze/większe”. Różnica pomiędzy $m = 0$, $M = 255$ a $m = 2$, $M = 250$ będzie niemal niewidoczna.



Rys. 1. Okno dialogowe POZIOMY przed i po korekcy ($m = 11$, $M = 236$).

Dzieje się tak dlatego, że jasność i wyjściowego piksela została zmieniona zgodnie ze wzorem

$$L(i) = \begin{cases} 0, & \text{dla } i < m, \\ \frac{255}{M-m}(i - m), & \text{dla } m \leq i \leq M, \\ 255, & \text{dla } i > M. \end{cases}$$

znów oczywiście po zaokrągleniu do całkowitej wartości.

Mówiąc krótko: najciemniejszy piksel naszego obrazu uzyskał obecnie wartość 0, najjaśniejszy – 255, a wartości pośrednie zostały odwzorowane liniowo na przedział $[0, 255]$. Wykorzystujemy teraz najciemniejszą i najjaśniejszą jasność pikseli, czyli pełną rozpiętość tonalną obrazu. Na ogół operacja ta istotnie polepsza wygląd zdjęcia (od tej reguły są oczywiście wyjątki), może się jednak okazać, że zdjęcie wydaje nam się nadal zbyt jasne, zbyt ciemne, za mało kontrastowe albo za bardzo kontrastowe.

Najpierw spróbujemy zaprząć do pracy matematykę zakodowaną w programie dość głęboko. Przeanalizujemy pewną szybką sztuczkę, polegającą na zastosowaniu warstw. Warstwy wykorzystuje się tak naprawdę do innego rodzaju edycji, a pokazana metoda jest w zasadzie ubocznym, choć interesującym zastosowaniem tej funkcjonalności. Dla uproszczenia wzorów wykonajmy jednak najpierw w myślach pewną operację – podzielmy wartości wszystkich pikseli przez 255 (piksel czarny będzie miał więc jasność 0, a biały 1), dopuśćmy też w myślach jasności kodowane wszystkimi liczbami rzeczywistymi z odcinka $[0, 1]$. Zastanówmy się teraz, co by się stało, gdyby każdy piksel o jasności $x \in [0, 1]$ otrzymał jasność x^2 ? Piksele czarne i białe pozostałyby czarne i białe (bo $0^2 = 0$ i $1^2 = 1$), ale pośrednie piksele uległyby przyciemnieniu (bo $x^2 < x$ dla $0 < x < 1$). Spróbujmy zrealizować ten pomysł.

Otwieramy nasze zdjęcie w GIMPie i w menu WARSTWY wybieramy opcję DUPLIKUJ WARSTWĘ (lub wciskamy CTRL+SHIFT+D). W oknie dialogowym WARSTWY (CTRL+L) zobaczymy dwie kopie obrazka, jedna nad drugą. Na razie widoczny obraz nie uległ żadnej zmianie. Kluczową dla nas opcją jest zmiana trybu górnej warstwy. Wynikowa wartość piksela wyliczana jest jako funkcja zależna od wartości piksela górnej i dolnej warstwy; oczywiście konkretna funkcja zależy od wybranego trybu. Domyślnie wybrany jest tryb ZWYKŁY (ang. *Normal*), w którym wartość wynikowa to po prostu jasność piksela górnej warstwy. Aby zrealizować nasz pomysł, zmieniamy tryb na MNOŻENIE (ang. *Multiply*) – ponieważ wartości pikseli górnej i dolnej warstwy są identyczne, w efekcie otrzymamy funkcję kwadratową, a obraz ulega zdecydowanemu przyciemnieniu.

No dobrze, a jak rozjaśnić obraz? Wystarczy przełączyć tryb górnej warstwy na PRZESIEWANIE (ang. *Screen*), w niektórych programach nazwany również mnożeniem odwrotności. Jeżeli wartości piksela górnej i dolnej warstwy wynoszą odpowiednio x i y , tryb ten przypisze wynikowemu pikselowi wartość

$$S(x, y) = 1 - (1 - x)(1 - y).$$

Dla $x = y$ otrzymujemy więc

$$S(x, x) = 1 - (1 - x)^2.$$

Nietrudno przekonać się, że jest to funkcja rosnąca, $S(x, x) \geq x$ oraz 0 i 1 są punktami stałymi $P(x, x)$, w efekcie uzyskamy więc rozjaśnienie obrazu bez strat spowodowanych nieróżnowartościowością funkcji, nie zostanie też ograniczona rozpiętość tonalna obrazu, nadal potencjalnie mamy szanse uzyskać wszystkie wartości pomiędzy 0 a 1.

Podobnymi metodami da się też wywołać wrażenie zwiększenia lub zmniejszenia kontrastu. Z pomocą przychodzi tryb POKRYWANIE (ang. *Overlay*), który działa zgodnie ze wzorem

$$O(x, y) = y(y + 2x(1 - y)),$$

gdzie x i y oznaczają wartości piksela odpowiednio górnej i dolnej warstwy. Dla $x = y$ otrzymujemy

$$O(x, x) = 3x^2 - 2x^3.$$

Jest to funkcja rosnąca w przedziale $[0, 1]$, pierwiastkami wielomianu $O(x, x) - x = -2x^3 + 3x^2 - x = -x(2x^2 - 3x + 1)$ są liczby 0, $\frac{1}{2}$, 1, w szczególności znów $O(0, 0) = 0$ i $O(1, 1) = 1$. Dokładniejsza analiza pokazuje, że $O(x, x) \leq x$ dla $0 \leq x \leq \frac{1}{2}$ oraz $O(x, x) \geq x$ dla $\frac{1}{2} \leq x \leq 1$. Oznacza to, że nasza funkcja

Aktualna w momencie pisania tego artykułu wersja GIMPa 2.6.8 ma w tym miejscu błąd i opcja Pokrywanie używa innego wzoru. Efekt jest jednak wystarczająco zbliżony do oczekiwanego.

przyciemnia piksele ciemne (tj. o wartościach nie większych od $\frac{1}{2}$) i rozjaśnia jasne (o wartościach nie mniejszych od $\frac{1}{2}$), a więc właśnie podnosi kontrast.

Jeżeli jednak górną warstwę zastąpimy jej negatywem (w GIMPie, przy wybranej górnej warstwie, KOLORY \rightarrow INWERSJA), nowa jasność piksela po zastosowaniu trybu pokrywania wynosić będzie

$$O(1 - y, y) = y(y + 2(1 - y)^2) = y(2y^2 - 3y + 2).$$

Uzależniamy tutaj wartość funkcji od y , czyli wartości piksela dolnej warstwy, bo to jest wartość oryginalnego obrazu, gdyż na górnej warstwie dokonaliśmy już inwersji, czyli przekształcenia $x \mapsto 1 - x$. Znów $0, \frac{1}{2}, 1$ są punktami stałymi tej funkcji, ale tym razem funkcja ta rozjaśnia ciemne piksele ($O(1 - y, y) \geq y$ dla $0 \leq y \leq \frac{1}{2}$) i przyciemnia jasne ($O(1 - y, y) \leq y$ dla $\frac{1}{2} \leq y \leq 1$), a więc zmniejsza kontrast.

GIMP oferuje nam jeszcze pośród wielu trybów mieszania warstw ciekawe tryby ROZJAŚNIANIE (ang. *Dodge*) i WYPALANIE (ang. *Burn*). Działają one odpowiednio zgodnie ze wzorami

$$D(x, y) = \frac{y}{1 - x}, \quad B(x, y) = 1 - \frac{1 - y}{x};$$

działanie i zbadanie destruktywności tych trybów pozostawimy Czytelnikowi do samodzielnego zbadania.

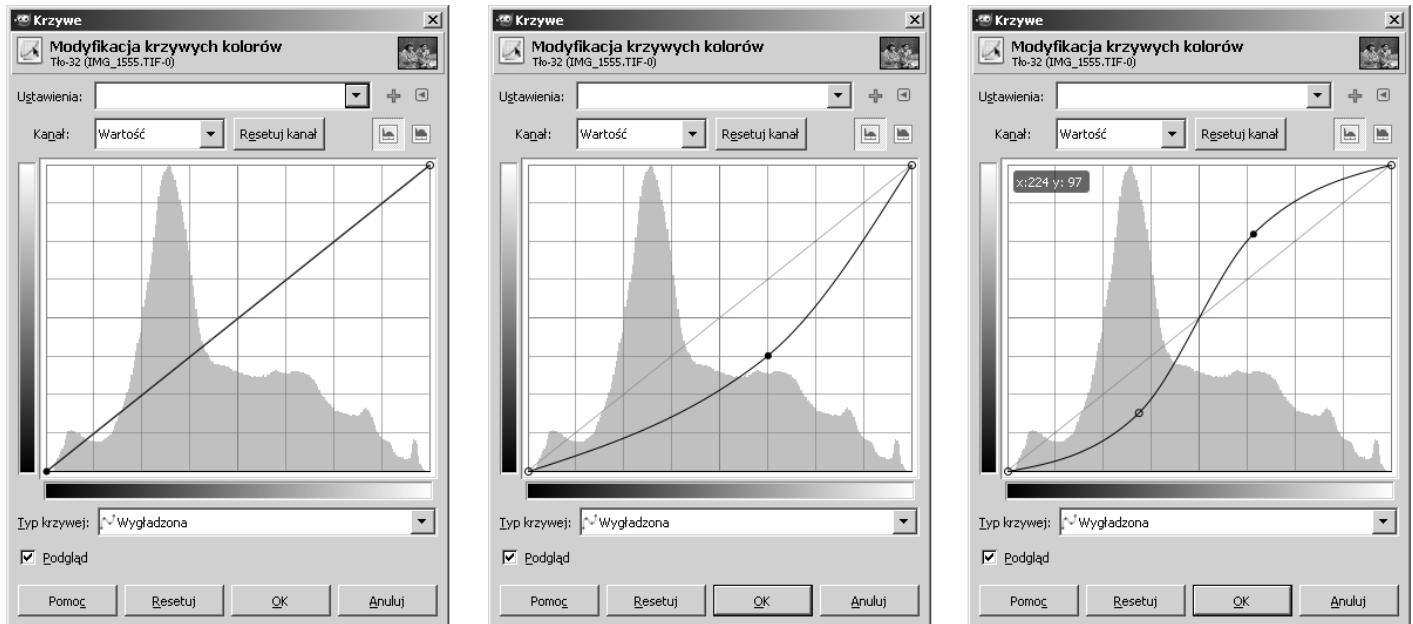
Odkrywszy, że na jasność pikseli można działać pewnymi funkcjami, chciałoby się móc na jasności pikseli oryginału zastosować dowolną funkcję ciągłą $f : [0, 1] \rightarrow [0, 1]$. Okazuje się, że istotnie można to wykonać i to bez bez pracy na warstwach! Z pomocą przychodzi nam narzędzie tzw. krzywych tonalnych (w GIMPie: KOLORY \rightarrow KRZYWE). Po otwarciu narzędzia ujrzymy wykres funkcji identycznościowej, który poprzez klikanie w kwadracie i przesuwanie punktów kontrolnych można dowolnie przekształcić do wykresu funkcji ciągłej z odcinka w odcinek. Nie jesteśmy więc ograniczeni tymi kilkoma wzorami zakodowanymi przez autorów programu jako tryby mieszania warstw, możemy dowolnie kształtować funkcję zmieniającą jasność pikseli. Funkcję taką definiujemy nie poprzez zadanie wzoru, ale kreśląc w narzędziu KRZYWE jej wykres – w praktyce jest to dużo wygodniejsze rozwiązanie.

W szczególności ciągnąc krzywą delikatnie w dół lub w górę możemy uzyskać bezstratne przyciemnienie lub rozjaśnienie obrazu, efekt będzie prawdopodobnie dużo subtelniejszy niż ten uzyskany przy pomocy funkcji kwadratowej. Kształtując krzywą w kształt zbliżony do litery S (zob. rys. 2) podnosimy subiektywne wrażenie kontrastu zdjęcia. Tu należy się jednak słowo wyjaśnienia: tak naprawdę kontrast zwiększany jest w tych obszarach zdjęcia, w których pochodna nakreślonej krzywej jest większa od 1. Krzywa S-kształtna zwiększa kontrast w tonach średnich, ale kosztem zmniejszenia kontrastu w światłach i cieniach. Wpływa to na subiektywny odbiór obrazu jako bardziej kontrastowego, niemożliwe jest jednak zwiększenie kontrastu we wszystkich obszarach, gdy zdjęcie wykorzystuje już pełną rozpiętość tonalną od 0 do 255.

Krzywą taką można kształtować niemal dowolnie aby uzyskać pożądany efekt, na przykład rozjaśniając tylko tony bardzo ciemne, albo zwiększając kontrast w tylko bardzo jasnych – ogranicza nas niemal wyłącznie nasza własna inwencja. Warto jednak pamiętać, by zawsze rysować funkcję niemalejącą, a najlepiej wyraźnie rosnącą, inaczej efekt nie jest specjalnie estetyczny.

W narzędziu krzywych tonalnych widzimy również histogram. Jeżeli uprzednio nie wykonaliśmy korekty poziomami, dobrym pomysłem jest poprowadzenie krzywej na poziomie 0 do pierwszego niezerowego punktu histogramu; analogicznie, jeżeli histogram pokazuje, że nie ma pikseli bardzo jasnych, takie wartości krzywa może bez strat odwzorować na 255 (pomiędzy tymi punktami kształtujemy ją dowolnie). W efekcie najciemniejszy punkt obrazu posyłany jest na 0, a najjaśniejszy na 255 – wykorzystujemy więc optymalnie rozpiętość tonalną obrazu cyfrowego.

Ze względu na zaokrąglenia do całkowitych wartości, funkcja działająca na dyskretnym zbiorze $\{0, \dots, 255\}$ na ogół nie jest różnowartościowa. Jednak w przeciwieństwie do opisanych na początku destruktywnych operacji wykonywanych za pomocą suwaków, jest to spowodowane wyłącznie zaokrągleniami, same zaś straty są w praktyce nieodczuwalne, zatem nazwanie tych operacji bezstratnymi jest zasadne.



Rys. 2. Narzędzie KRZYWE, od lewej kolejno: bez modyfikacji, krzywa przyciemniająca i krzywa zwiększająca kontrast

Właściwe wykorzystywanie krzywych tonalnych wymaga dobrej intuicji, w jaki sposób funkcja o nakreślonym przez nas wykresie wpłynie na jasności pikseli i kontrasty – matematyk ma w tym miejscu dużą przewagę. Powodzenia w eksperymentach!