

Jau trīs gadus mēs strādājam poligrāfijas tirgū un sniedzam tehnoloģiskās konsultācijas dažādos drukas veidos strādājošām tipogrāfijām, reklāmas aģentūrām, izdevniecībām un visiem citiem poligrāfijas jomā iesaistītiem uzņēmumiem. Veidojam arī attiecības ar speciālistiem no citām valstīm. Viens no pazīstamākajiem Ukrainas tehnologiem Aleksandrs Piļskis ir devis atļauju publicēt viņa rakstu. Cerēsim, ka šis zinātniski pētnieciskais darbs atvieglēs jūsu ikdienas darbu.

Jeļena Kuzmina
SIA “Poligrāfijas
tehnoloģiju konsultācijas”

Trapping un overprint

Šis raksts ir paredzēts sagatavotiem lasītājiem. Tiem, kas ir saskārušies ar krāsu sakrītību pilnkrāsu drukas gadījumos un krāsu nesakrītības kļūdām, tā saucamajām **misregistracion**¹. Un arī tiem, kas interesējas par šīs problēmas² risināšanas paņēmieniem.

¹ *Misregistration* pseido kontūru paaugstināta redzamība ir saistīta ar cilvēka acs (precīzāk smadzeņu) pieradumu (kas nāk no bērnības) jebkurā attēlā, pirmām kārtām, meklēt informāciju. Pat tur, kur tai nav jābūt. Nesakrītības defekti veido pamanāmus parazītiskus elementus (artefaktus), kas pasliktina kopējo attēla uztveršanu. Īpašs jutīgums acij ir pret krasām košuma izmaiņām, defekti krāsu toņos ir krietni mazāk pamanāmi. Košuma artefaktu nomainīšana uz defektiem krāsu toņos arī ir mūsu *overprint* un *trapping* maskēšanas paņēmieni pamatā.

² Pat ideāla iespiedmašīna nevar pilnīgi precīzi savietot krāsas. Galvenais iemesls – papīra deformācija, kas vienmēr notiek, papīram virzoties caur iespiedmašīnu. Praksē precizēt gaidāmo nesakrītību labāk pie tipogrāfijas tehnologa. Parasti par normālu skaitās nesakrītība rastra puslīnijas platumā. Piemēram, 175 lpi liniatūrai – 70 mkm.

Prasīt ideālu sakrītību no iespiedēja ir bezjēdzīgi. Cenšoties panākt sakrītību “uz nulli”, var tikai sabojāt papīru.

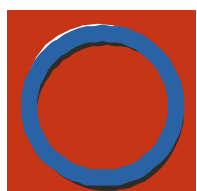
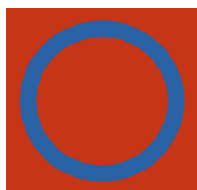
Par terminiem

Ir divas iespējas izvairīties no *misregistration* problēmas. Pirmā ir vienkārša – veidot *overprint*, bet otra sarežģītāka – *trapping* (lietosim šos terminus oriģinālajā angļiskajā rakstībā). Pirmais veids nozīmē vienas krāsas uz otras uzklāšanas režīma vadīšanu. Otrais veids, – uz speciālu objektu – *mistreregistration* lamatu – izmantošanas pamata. Abi veidi ir cieši saistīti un parasti tiek lietoti kopā.

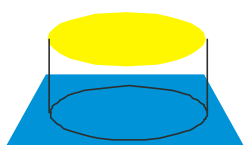
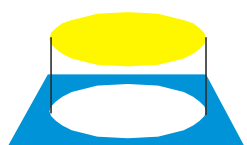
Sāksim ar vienkāršu piemēru. Mēs gribam nodrukāt dažādu krāsu objektu pāri. Vienu objektu virs otra. Kas notiek tādā gadījumā? Augšējais objekts var “izcirst caurumu” apakšējā – t.i. izdarīt **knockout**. Iegūtais iedobums – speciāli izveidots trešais objekts, augšējā objekta projekcija uz apakšējā. Tikko apakšējā objektā parādās iedobums, uzreiz arī rodas problēma – drukas laikā ar augšējo objektu precīzi jātrāpa apakšējā objektā izveidotā “caurumā”. Jāsavieto tie. Bet vai vienmēr ir nepieciešama projekcijas veidošana?

1 daļa. Overprint

Daļā gadījumu mēs varam aizliegt objektam veidot iedobumu zem sevis. Tam arī ir domāts speciāls atribūts – *overprint*. Objekts ar atribūtu *overprint* neveido projekciju. Tā rezultātā nenāksies savietot objektu un iedobumu (projekcijas nebūs). Nav “cauruma” – nav arī problēmu ar savietošanu.



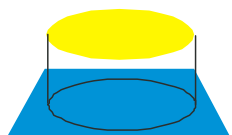
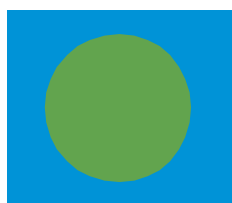
1. zīm. Nesakrītība pie drukas



iedobums - *knockout*

uzklāšana - *overprint*

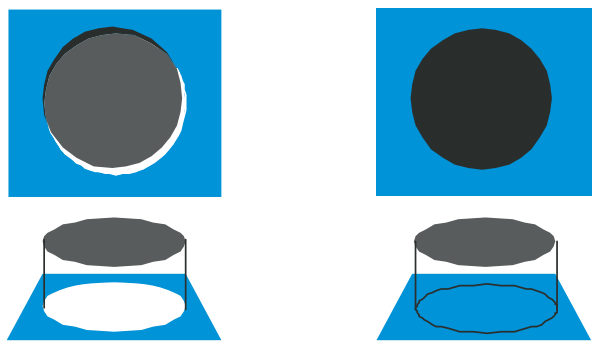
2. zīm. Divi drukas varianti



3. zīm. Krāsu nobīdes uzklāšanas laikā

Kad drīkst, bet kad nedrīkst lietot *overprint*? Padomāsim – kam mums vispār bija vajadzīga projekcija? Lai izvairītos no vienas krāsas uzklāšanas uz otras. Bet kāpēc? Lai izvairītos no krāsu nobīdes. Parastās ofseta krāsas taču ir caurspīdīgas. Ja dzeltenais objekts ir uz gaiši zila fona bez iedobuma tajā, uzklājot vienu krāsu virs otras, būs divu krāsu summa un objekts kļūs zaļš (3. zīm.). Vēl ļaunāk, ja par fonu kalpos raibs attēls (rastra vai vektoru – nav būtiski). Attēls atklāti “spīdēs cauri” objektam. Bet ja uztaisīt *overprint* baltam objektam? Tas vienkārši pazudīs. Tāpēc parasti izdevniecības programmās visiem objektiem pēc noklusēšanas tiek piešķirts *knockout* atribūts (tiek veidotas objektu projekcijas).

Overprint Black



4. zīm. Melnā krāsa – *knockout* un *overprint*



5. zīm. Lielas burtu bieznes gadījumā *overprint* kļūs redzams

Kādos gadījumos *overprint* ir pieņemams? Kad krāsu nobīdes nav bīstamas? Kad tās nav pamanāmas. Piemēram, ja objekts ir melns un sīks (4. zīm.). Ja melnais teksts 8 punktu lielumā ir izvietots uz gaiši zila fona ar atribūtu *overprint*, nobīde K+C krāsu pārklāšanās vietā nav būtiska. Nobīde ir (teksts paliks nedaudz tumšāks), bet tā praktiski nav pamanāma. Bet, ja mēs aizmirsīsim par *overprint*, jebkura skatītāja acs manīs visniecīgāko nesakrītību starp tekstu un iedobumu zem tā gaiši zilā fonā (kreisais zīmējums). Runājot kopumā, izvēles jēga ir vienkārša – jāizlemj, kas ir vairāk pamanāms – nesakrītības defekti vai

neprecizitātes krāsu toņos un “caurspīdīgums”. Melniem objektiem *misregistration* ir daudz vairāk manāma, nekā nesakrītība. Tāpēc par parastu praksi pirmsdrukas sagatavošanas iecirknī ir kļuvusi melnās krāsas iestatīšana uz pārklāšanos – ***all black overprint***.

Bet šis iestatījums ir bīstams.

Tikai sīkiem melniem objektiem var ignorēt krāsu nobīdes un “caurspīdīgumu”. Pasvītāju – tikai sīkiem objektiem. Ofseta melnā krāsa ir tumša, bet nav absolūti melna. Un pie tam vēl pietiekami caurspīdīga. Ja jūs iestatīsit *overprint* melnās krāsas segumam ar laukumu 5x5 cm un izvietotam virs tā attēlam arī, attēls “spīdēs cauri” melnajam laukumam. Ja jūsu līnija ir 5

mm bieza, *overprint* tai iestatīt nav vēlams, jo objekti, kas atrodas zem tās, būs viegli nolasāmi ar aci. Analogiski notiek arī liela izmēra burtiem (5. zīm.). Tātad parastā pirmsdrukas sagatavošanas iecirkņa prakse “vienmēr melnās *overprint*” var arī pievilt. Melnās krāsas diferencētai *overprint* atribūta iestatīšanai ir nepieciešams intelekts:

“teksts ar burtu lielumu ne vairāk par...”

“*stroke* ar biezumu ne vairāk par...”

“*fill* izmērā ne vairāk par...”

Overprint atribūts var būt iestatīts tieši maketā, bet var arī reprocentra rastra procesorā (RIP). Kur ērtāk? Atbilde ir vienkārša – tur, kur ir speciālists – cilvēks (vai “mākslīgais intelekts”), kas zina iepriekš minētos, visvienkāršākos pamata noteikumus. No abām pusēm ir bīstama varbūtība nospiegt “slinko” pogu ***all black overprint***. Noteikti jāatceras – pat kvalitatīvs *overprint* nevar aizstāt *trapping* iestatījumu. Ja jūs nevarat uztaisīt (pasūtīt) *trapping* – centieties izmantot primitīvākus *overprint* iestatījumus, pat uz “caurspīdīguma” rēķina. Pilnvērtīgā (*overprint* + *trapping*) iestatījumu veidošanas variantā es iesaku sekojošus iestatījumus priekš ***black overprint***:

- teksta burtu lielums nepārsniedz 12 pt;
- *stroke* un *fill* ne vairāk par 1 mm (~3 pt).

Ja tā nebūs augstas klases produkcija un *trapping* netiks izpildīts, ir vērts palielināt šos skaitļus:

- teksta burtu lielums nepārsniedz 24 pt;
- *stroke* un *fill* ne vairāk par 2 mm (~6 pt).

Var būt pieņemamas arī lielākas vērtības, bet tas jau ir atkarīgs no attēla, un lēmuma pieņemšanai jāpieiet radoši. Tikai noteikti jāsaprot, ka *misregistration* parādīšanās ir daudz bīstamāka par attēla

“caurspīdīgumu” caur melnās krāsas klājumu. Reizēm ir labāk atstāt *all black overprint* – būs “trešā šķira, nevis brāķis”.

Sedzošās krāsas

Kādos vēl gadījumos krāsu nobīdes pārklājumā nav bīstamas? Izskatīsim necaurspīdīgu, tā saucamo, sedzošo (*opaque*) krāsu pielietošanas variantu. Atšķirībā no ofseta (caurspīdīgām), sedzošās krāsas nespīd cauri. Ja mūsu piemērā (2. zīm.) drukai tika pielietota speciālā sedzošā dzeltenā krāsa (un tā tika uzklāta pēc gaiši zilās), tad krāsu summa pārklāšanās vietās neveidojas. Un tādā gadījumā, atšķirībā no parastām krāsām, ir ļoti būtiska krāsu uzklāšanas secība – dzeltenā krāsa jāuzklāj pēc gaiši zilās.

Sedzošas var būt ne tikai krāsas. Skārds – lielisks sedzošā materiāla piemērs. Ja nopietni, tad par sedzošām poligrāfijā tiek uzskatītas folijas, kas tiek izmantotas spiedumam, daudzas trafareta krāsas. Mēdz būt arī sedzošās ofseta krāsas, piemēram, sudrabs, bronza un citas (tā saucamie “metāliki”). Sedzošas, bet ar piezīmi. Plakanā ofseta drukas veidā diemžēl nav iespējams uzklāt pietiekami biezu krāsas slāni. Bet gadījumā, ja krāsas slānis ir 1-2 mkm biezs, sudrabam citas krāsas nedaudz spīd cauri, tāpēc tādas krāsas labāk nosaukt par pussedzošām (*semiopaque*)³. Diezgan bieži sedzošās krāsas sajauc ar citām, caurspīdīgām. Vai dzidrina (“izbalina”), pielietojot caurspīdīgas lakas vai arī, izmantojot rastru. Tādos gadījumos sedzošās krāsas īpašības samazinās un *overprint* izmantošana var novest pie kļūdām.

³ visām krāsām, neatkarīgi no tā, vai tās ir ofseta, trafareta vai cita tipa, ir dažāda caurspīdīguma pakāpe (precīzāk, sedzošās spējas). Sedzošās spējas līmeni apraksta speciāls parametrs – *opacity*. Bieži, lai nesarežģītu tēmu, krāsas sadala četrās kategorijās – caurspīdīgās, puscaurspīdīgās, pussedzošās, sedzošās. Parastās triādas krāsas (tā saucamās *process*) – ir tipiskas *semi-transparent* (puscaurspīdīgās).

Overprint atribūta izmantošana sedzošām krāsām pamatā tiek noteikta pēc to secības kārtības (*print order*)⁴, bet var būt atkarīga arī no citām tehnoloģiskajām niansēm un ierobežojumiem. Atklāt tos visus nav iespējams. Tikai viens piemērs – iepriekš minētā folijas spieduma gadījumā nesakritības kļūdas ir diezgan lielas. Folija ir labi sedzošs materiāls, tāpēc klišejas zīmējumam maketā parasti piešķir *overprint* atribūtu. Bet nedrīkst aizmirst par attēlu zem folijas (precīzāk, par krāsas slāņa biezumu). Ja spiedums ir liels un tiek uzlikts virs tumša zīmējuma, zem spieduma būs daudz krāsas. Krāsa var būt arī nepietiekami nožāvēta. Tādos gadījumos spieduma laikā folija var neuzklāties zīmējumam vispār vai arī mest burbuļus. Kā redzam, situācija ir līdzīga *overprint black*, kaut gan iemesli ir pavisam dažādi. Šajā gadījumā ir nepieciešams pieņemt lēmumu – kad var iztikt ar *overprint*, bet kad jāpielieto *trapping*.

⁴ Parastās (*process*, vai kā pieņemts *trapping* terminoloģijā – *normal*) krāsas nav ideāli caurspīdīgas, tāpēc ir jāņem vērā arī šo krāsu uzklāšanas secība. Bet mūsu maskēšanas uzdevumiem tas nav būtiski. Tas ir vairāk aktuāli krāsu dalīšanai, kur ir labi zināms, ka M+Y un Y+M – nav viens un tas pats. Abas ir sarkanas, bet viena “pionieru”, bet otra aveņu. Tāpēc ofsetā (densitometrijā) divu krāsu uzklāšanas parametrs (binārs) tiek speciāli novērots. Šo parametru arī sauc par *trapping*, bet tas ir tikai nosacīti attiecināms uz lamatu uzbūves procesu.

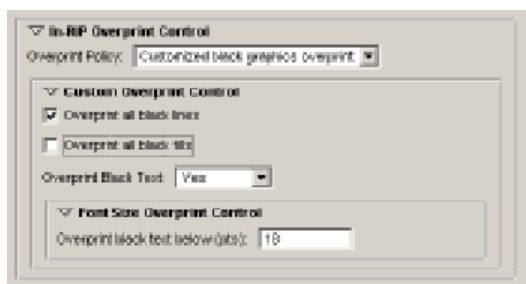
Pielietojuma prakse

Diezgan bieži maketos ir sastopami ne tikai attēli, bet arī izciršanas/bigošanas kontūras. Tādām kontūrām viennozīmīgi pielietojam *overprint*. Analogiski arī ar fragmentārās lakošanas formām un ar “akla” spieduma klišejas attēliem. Vēl viens piemērs – sedzoša balta pamatne krāsaino vai caurspīdīgo materiālu apdruckas gadījumos (6. zīm). Tādai pamatnei jābūt nokrāsotai speciāli iezīmētā krāsā (*spot color*) *opaque white*, visās izdevniecības programmās iebūvētā *white* krāsa šajā gadījumā neder.

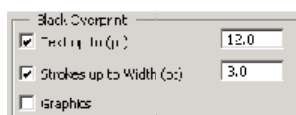
Dažas piezīmes maketu izvada speciālistam. Diemžēl ar rastra procesora līdzekļiem pagaidām nav iespējams pilnībā automatizēt *overprint* atribūta vadīšanas procesu. Rastra



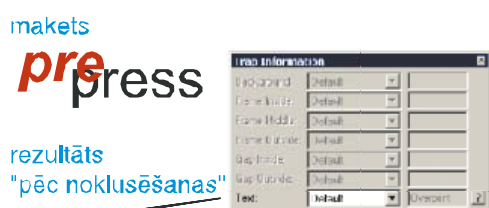
6. zīm. Sedzošā baltā krāsa uz caurspīdīgas pamatnes



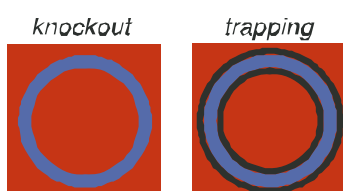
7. zīm. Black overprint Agfa Apogee procesorā



8. zīm. Heidelberg Supertrap 4.0.



9. zīm. Kļūda "teksts pa tekstu" QuarkXpress programmā



10. zīm. Krāsu lamatas

Nesakritības defektu manāmību ērti novērtēt melnbaltajā režīmā (11. zīm.).

Nedaudz terminoloģijas

Papildus vienkāršam *knockout* ir vēl divi projekcijas veidošanas varianti (12. zīm.) Objekts ir lielāks, nekā viņa projekcija, vai otrādi, mazāks par to. Abi varianti tiek lietoti *trapping* gadījumā. Kādu izvēlēties – atkarīgs no tā, kurš ir mazāk pamanāms. Vislabākais *trapping* ir nepamanāms.

- *spread* – objekta paplašināšana attiecībā pret projekciju;

procesoru daļai ir iespēja analizēt objektus uzstādot *overprint* priekš melnās krāsas (7. un 8. zīm.). Bet šīs iespējas nav pilnvērtīgas. Ir vajadzība uzstādīt *overprint* ne tikai melnai krāsai, bet arī citām. Izdarīt to automātiski krāsai ar nosaukumu "izciršana" (vai līdzīgu) var tikai *Creo-Scitex* firmas rastra procesori. Pārējos gadījumos *overprint* atribūtu nāksies uzstādīt pielikumā vai PDF failā ar roku.

Atcerieties par baltās krāsas *overprint* kontroli.

Daudzas izdevniecības programmas var uzstādīt *overprint* tādai krāsai kā *black 0% tint*. Tas ir baltai. Jebkurai *preflight* programmai obligāti jāpārbauda *always white knockout*. Protams, izslēdzot situāciju ar sedzošo balto krāsu.

Teksta *overprint* programmā *QuarkXpress* jāpievērš īpaša uzmanība. Šī programma cenšas uzstādīt *overprint* atribūtu jebkuram (!) objektam, kas atrodas virs baltā (*white underneath color*). Reizēm šī riskantā metode izsauc kļūmi. Tā kā *QuarkXpress* vienmēr nosaka teksta bloka krāsu pēc tā fona, "teksts pa tekstu" gadījumā, (9. zīm.) sarkanais teksts, kas ir izvietots melnajam pa virsu, kļūdaini tiek noteikts kā izvietots pa virsu baltajam un gūst atribūtu *overprint*.

Īpaši atzīmēšu – neaizmirstiet sekot līdz *overprint* atribūtam, pārtaisot maketu, īpaši, mainot krāsas.

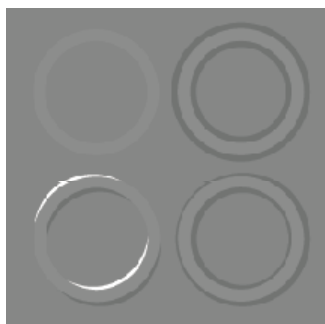
2.daļa. Trapping pamati

Gadījumos, kad nedrīkst izmantot *overprint* atribūtu, nesakritības defektu maskēšanai tiek pielietots otrais veids – *trapping*.

Vārds *trap* tulkojumā nozīmē "lamatas". *Prepress* kontekstā – krāsu lamatas *misregistration* defektiem. 10.zīmējumā ir parādīts, kā speciālas lamatas slēpj nesakritību starp zilo riņķi un izciršanas štanci priekš tā sarkanajā fonā.

Pēc kāda principa strādā lamatas? *Knockout* režīmā objekts un iedobums zem tā ir precīzi vienādi. Pat sīkākā nesaskaņotība – un veidojas acīm redzami baltie misēkļi. Bet, ja mēs nedaudz palielināsim zilā apla biezumu, bet iedobuma izmērus nemainīsim, zilajam aplim parādīsies tumši violeta kontūra, tā saucamais *keyline* (augšējā zīmējuma labajā pusē). Liekas, ka tas ir slikti, bet apskatieties divus apakšējos zīmējumus, kur ir imitēta krāsu nesakritība. *Knockout* režīmā (apakšējā zīmējuma kreisajā pusē) zīmējumam ir parādījušies gan baltie, gan tumši violetie parazitējošie elementi. Lielie košuma lēcieni. Labajā zīmējuma pusē īpašas problēmas nav redzamas. Bet radušās kontūras tikai izceļ zilo apli, palielina attēla skaidrību, izveidojot *unsharp mask* līdzīgu efektu. Tādā veidā, nedaudz nobīdot krāsas (zila ar violetu kontūru), mēs nomaskējam krasus košuma lēcienus. Šādā paņēmienā arī slēpjas *trapping* jēga.

Nesakritības defektu manāmību ērti novērtēt melnbaltajā režīmā (11. zīm.).



11. zīm. Vērtējums pēc pelēkā komponenta

- *choke* – projekcijas saspiešana, samazināšana attiecībā pret objektu;
- *keyline* – artefakts, *trapping* elementa parādīšanās rezultāts;
- *ND (Neutral Density)* – krāsas košums, precīzāk, tās ahromatiskā sastāvdaļa (pelēka vai neitrāla)⁵.

Ar pēdējo terminu detalizēti tiksim skaidrībā ar tipisku puscaurspīdīgu krāsu izmantošanas piemēra palīdzību.

⁵ *ND* parametrs pēc savas būtības ir līdzīgs vispāratzītai krāsas košuma definīcijai (*luminosity*). Tāpēc ļausim sev šajā rakstā vietām izmantot terminus “košums” un “gaišums”.

Kā mēs redzam, *trapping* ievieš maketā grozījumus. Pēc *trapping* atribūta piešķiršanas uz objektu robežām rodas blakus kontūras. Tās ir

keyline. Mūsu uzdevums – padarīt tās neredzamas. Kā? Dizaineri labi pazīst *contour-defining* principu, pēc kura objekta forma tiek noteikta ar tā kontūru. Tumšajam objektam uz gaiša fona vizuālo izmēru norobežo (“zīmē”) tumšā kontūra. Gaišajam objektam uz tumšā fona – otrādi. Uz šo noteikumu pamata precīzējam uzdevumu – ar *trapping* atribūta piešķiršanu saistīto attēlu deformācijas minimizācijai, katram objektam nepieciešams saglabāt atbilstošas kontūras izmērus. Mūsu *keyline* vienmēr tumšs, tāpēc tam vienmēr jāatrodas tumšajā objekta pusē. Tādā veidā tiek pieņemts lēmums, kurā virzienā veidot *trapping*, kad pielietot *spread*, bet kad *choke* (12. zīm.).

Tātad, pirmais *trapping* noteikums:

Gaišais objekts vienmēr tiecas zem tumšā.

Bet kura no mūsu krāsām ir gaišāka-tumšāka? Un par cik? Daudziem ir pazīstams tāds krāsas parametrs, kā blīvums (*D*). Jo lielāks blīvums, jo intensīvāka, “stiprāka” krāsa. Bet diemžēl ar blīvuma palīdzību nevar salīdzināt krāsas. Piemēram, skaidrs, ka dzeltenā krāsa ir gaišāka par melno. Bet izlemt, kura ir tumšāka – *magenta* vai *cyan*, ir daudz grūtāk. Tāpēc tika ieviests īpašs parametrs – neitrālais blīvums (*Neutral Density* vai *ND*). Šis parametrs speciāli izdala ahromatisko krāsas komponentu, kas ļauj salīdzināt “stiprumu”, dažādu krāsu objektu gaišumu. *ND* vērtība arī nosaka, kurš no objektiem ir gaišāks, bet kurš tumšāks.

Ir labi zināms, ka poligrāfijā krāsa tiek noteikta, izmantojot krāsu dalīšanu, t.i. vairāku krāsu sintēzi. Tāpēc objektu gaišuma analīze jāizmanto tikai atsevišķām, drukai izmantojamām krāsām. Visbiežāk izmanto *CMYK* sistēmu, kas satur četras bāzes krāsas. Toties bieži sastopama arī citu krāsu pielietošana (tā saucamās *spot color*, piemēram, labi pazīstamās *Pantone* sistēmas krāsas). Visas sintēzes laikā izmantotās krāsas turpmāk saucim par šīs krāsas komponentiem, bet krāsas, kas satur vairāk par vienu komponentu – par saliktām. Vistumšākā *CMYK* krāsa (komponents) – melnā. Tā ir ahromatiska, tāpēc melnai krāsai $ND_{Black} = D_{Black}$. Ar citām krāsām ir sarežģītāk. *Trapping*-sistēmās *ND* datu uzglabāšanai tiek pielietotas speciālas tabulas, kas satur ne tikai tipiskas populāro triādu vērtības (1. tabula), bet arī *Pantone* bibliotēku detalizētas krāsu tabulas (*color tables*). Iespējama arī *ND* aprēķināšana no *CMYK* vai *Lab* krāsu koordinātēm, bet tāda paņēmiena precizitāte ir zemāka. Ar vidējo (no minētajām tabulām) datu izmantošanu vai *ND* aprēķināšanu no *CMYK* saistītie precizitātes zudumi praksē ir pilnīgi pieņemami. Īpašos gadījumos ir nepieciešams izmantot densitometrus, ar kuru palīdzību var precīzi nomērīt *ND* (*V – visual režīms*).

1. tabula. *Neutral Density* divās populārās triādās

ND	Cyan	Magenta	Yellow	Black
Euro	0.51	0.62	0.04	1.67
SWOP	0.60	0.76	0.16	1.73

Krāsas košumu rastrā var aprēķināt pēc formulas:

$$ND = -ND_{black} \lg(1 - R(1 - 10^{-ND_{color}/ND_{black}}))$$

kur ND_{color} – atbilstošas krāsas pilna seguma 100% *Neutral Density*;

R – rastra vērtība.

Neutral Density parametrs – neitrālā komponenta līmeņa decimālais logaritms. Tas ir parocīgs ar to, ka saliktās krāsas ND – ir vienkārši ND komponentu⁶ aritmētiskā summa.

⁶ Teorētiski tas neatbilst patiesībai, bet praksē 90% gadījumu ar tāda pieņēmuma precizitāti pilnīgi pietiek.

Divi trapping režīmi

Pastāv divas dažādas situācijas, kurās ir nepieciešams veidot *trapping*. Iepriekš izskatītie piemēri, kuros piedalījās divi krāsainie objekti un *trapping* veidojās uz to robežas – tas ir, tā saucamais “mitrais trapings” (*wet trap*)⁷. Bet *trapping* var būt vajadzīgs arī citos gadījumos. Piemēram, divu vai trīs komponentu objekts uz baltā fona, vai otrādi, – tiek pielietots inversais objekts uz saliktā fona. Tādā gadījumā jārunā par individuālā objekta *trapping* (precīzāk, krāsa-papīrs, pāra *trapping*). Tādu režīmu sauc par “sauso trapingu” (*dry trap*). Abiem režīmiem ir daudz kopīga, bet ir arī atšķirības, tāpēc izskatīsim tos atsevišķi.

⁷ Praktiski visas automatizētās *trapping* sistēmas strādā *wet trap* režīmā, otrajam režīmam pielietojot tikai ievilkšanu zem supermelnās (*choke rich black*) krāsas.

Krāsu pāra trapping

Pirms *trapping* elementa veidošanas mums jāatbild uz jautājumu – vai tas ir vajadzīgs. Izņemot tās situācijas, kad pietiek ar *overprint* (sk. 1. daļu), ir arī tādas, kur var pielietot vienkāršu *knockout*. *Trapping* ir vajadzīgs:

1. Ja viena no krāsām pārī ir vienkāršs otrās krāsas atvasinājums.

Pilnīgi skaidrs, ka starp objektiem ar krāsām **C100** un **C50** nesakritības problēmu nebūs. Protams, ja abi objekti tiek drukāti no vienas iespaidformas. Tādos objektos nav iedobuma, tāpēc arī nav artefaktu parādīšanās riska. Analogiski ir arī ar saliktām krāsām. Piemēram, krāsa **C50M50Y50** ir **C70M80Y100** krāsas atvasinājums (2. tabula) – tādā pārī taisīt *trapping* ir bezjēdzīgi. Otrā krāsa pilnībā sastāv no pirmās komponentiem, tāpēc iedobums tādā pārī arī nerodas. Pieņemts teikt, ka tādā gadījumā pirmā krāsa ir otrās krāsas atvasinājums. Atzīmēsim, ka krāsas var krietni atšķirties savā starpā, bet no *trapping* viedokļa tās ir “radnieciskas”.

2. Ja krāsu starpība pārī nepārsniedz *color step limit*

Kā noteikt, vai saliktās daudzkomponentu krāsas pārī ir “radnieciskas”? Precizēsim pirmo noteikumu. Izskatīsim katru komponentu (krāsu) pārī atsevišķi. Ja visi otrā objekta komponenti ir mazāki nekā pirmā, tad tie ir acīmredzami “radnieki”. Un ja nē, bet atšķirība ļoti nosacīta? Robežu, uz kuras starpība skaitīsies nosacīta, pieņemts saukt par *color step limit*. Tā ir regulējama. Jo šī robeža ir zemāka, jo vairāk *trap* objektu (*keyline*) mums ir jāizveido, jo vairāk deformāciju ieviešam sākotnējā maketā. Parastai komercprodukcijai tipiska ir 25% *step limit* vērtība. Ar vienu svarīgu nosacījumu.

2. tabula. Analīze pēc komponentiem

	Color 1	Color 2	Starpība	Trapping
C	70%	50%	20%	→
M	80%	30%	50%	→
Y	100%	60%	40%	→
K	0%	0%	0%	nav vajadzīgs
Tātad: visi komponenti vienā virzienā				nav vajadzīgs

3. tabulā mēs redzam, ka komponentu atšķirības ir pretējos virzienos, tie nav “radnieciski”. Visām krāsainajām sastāvdaļām starpība it kā nav liela – 10%, kas ir mazāk par tipisku *color step limit*. Bet, ja

dzeltenai krāsai 50-60% kritums ir praktiski nepamanāms un *trapping* nav vajadzīgs, tad zilās krāsas kritumam par 5-15% jau ir nepieciešama maskēšana. Mūsu *step limit* parametrs nav precīzs. Tāpēc mūsdienu *trapping* sistēmās tiek pielietota nosacītās krāsu starpības aprēķināšanas metode, tā saucamā **relative color step limit**. Tā tiek noteikta kā starpība, dalīta ar mazāko no krāsu vērtībām. Bet gadījumā, ja absolūtā starpība ir mazāka par 5%, nosacītā vērtība netiek aprēķināta un šis komponents netiek izskatīts. Tipiska *relative* metodes vērtība – 200%.

3. tabula. Absolūtā un nosacītā krāsu starpība

	Color 1	Color 2	Starpība	Nosacīta starpība	Trapping
C	15%	5%	10%	10/5=200%	←
M	25%	35%	10%	10/25=40%	→
Y	50%	60%	10%	10/50=20%	nav vajadzīgs
K	15%	10%	5%	netiek aprēķināts	nav vajadzīgs

Tātad: ir divi komponenti, kuru nosacītā starpība ir lielāka par *relative step limit* un tie ir pretējos virzienos

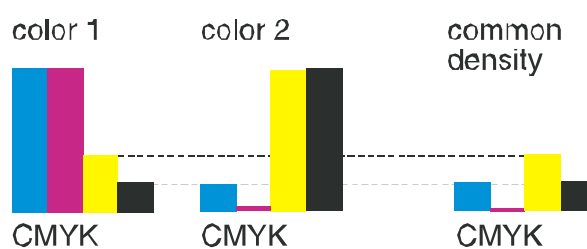
vajadzīgs

Apkoposim abus noteikumus: uzskatīsim divus komponentus par vienādiem, ja viena komponenta vērtības nepārsniedz otrā komponenta vērtības par *relative step limit* lielumu.

Trapping starp divām krāsām nav vajadzīgs, ja visi vienas krāsas komponenti ir lielāki vai vienādi (ņemot vērā *relative step limit*) ar otrās krāsas atbilstošajiem komponentiem.

3. Pāris sasniedz *common density limit*

Objektu pāra komponentu analīzes laikā ir nepieciešams arī ņemt vērā to kopējo (*common*) sastāvdaļu. Tā nosaka projekcijas krāsu, kas veidojas, uzklājot vienu objektu virs otra. 14. zīmējumā



13. zīm. Krāsu pāra kopējais komponents

parādīts piemērs – krāsu pāris **C100M100Y40K20** un **C20M0Y100K100**. Tam ir kopējā sastāvdaļa **C30M0Y40K20**. Šī sastāvdaļa arī ir visgaišākā krāsa, kas var rasties pie pāra nesakritības. Ja kopējā sastāvdaļa ir pietiekami tumša, *trapping* tādām pārim nav vajadzīgs. Augstas kvalitātes produkcijai ir rekomendēts turēt tipisku robežu (*common density limit*) 0,5D līmenī, vienkāršiem darbiem (ofseta-avīžu) to var pazemināt līdz 0,35-0,4D.

4. Kaut vai viena krāsa pāri ir ļoti gaiša

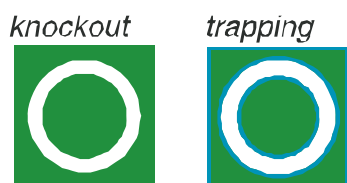
Ja viens no objektiem mūsu pāri pēc sava košuma (neitrālās komponentes) nenozīmīgi atšķiras no tīra papīra, *trapping* tādām pārim noteikti nav vajadzīgs. Maksimālā artefaktu košumu starpība, kas rodas nesakritības rezultātā, nedrīkst pārsniegt gaišākā objekta *ND*. Tāpēc, ja vienas krāsas košums pāri nepārsniedz 0,04 *ND* (par atskaites punktu tiek ņemta dzeltenā), pilnīgi pietiks ar parasto *knockout*⁸. Diezgan bieži tādās situācijās var tikt pielietots tā saucamais *inverse trapping*⁹ – nevis maskēt nesakritību ar *choke-spread* palīdzību, bet otrādi – bīdīt krāsas savā starpā. Piemēram, pāri **C100 – Y100** dzelteno krāsu būtu labāk atdalīt no zilās, veidojot baltu kontūru starp tām (*white outline*). Baltās kontūras redzamība tādā pāri ir zemāka par nesakritības rezultātā parādījušās zaļās krāsas pamanāmību.

⁸ Tādā situācijā sāk darboties citi noteikumi, jo tāds režīms ir līdzīgs krāsa-papīrs pāra *trapping* (*dry trap*) noteikumiem.

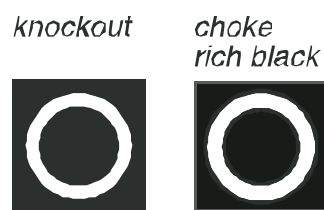
⁹ Piemēram, *Heidelberg Supertrap* programmā tāds režīms ir nosaukts par *keepaway mode*.

Vienas krāsas *trapping*

Nesakritību maskēšana tādā režīmā ir sarežģītāks uzdevums. Pirmajā plānā iznāk krāsu kļūmes *misregistration* dēļ. Brīžiem nav iespējams izlemt, kā labāk noslēpt nesakritības artefaktus krāsainajā trīskomponentu tekstā, kas ir izvietots uz balta papīra. 14. zīmējumā ir parādīts piemērs, cik ļoti kļūst redzami krāsu artefakti uz balta papīra fona. Mēs redzam: kaut gan izdarītais dzeltenās krāsas *choke* ļāvis



ideāla sakritība nesakritība
14. zīm. Krāsu artefakti



ideāla sakritība



nesakritība

15. zīm. Ievilkšana zem melnās

nedaudz samazināt nesakritības redzamību, zilās krāsas artefakti palikuši diezgan pamanāmi. Problēmas būtība: maskēšanas pamatpaņēmieni pārstāj darboties – košuma artefaktu nomaiņa uz krāsu artefaktiem. Baltais fons izceļ pat mazākās nesakritības kļūmes. Tāpēc vēlams izvairīties no tādiem problemātiskiem elementiem, kā divu un trīs komponentu maza izmēra burtiem, tievām līnijām, inversiem tekstiem uz sarežģīta krāsaina fona un tamlīdzīgiem, vēl dizaina izstrādes stadijā. Ja no sarežģītu (piemēram, “firmas”) krāsu izmantošanas izvairīties nav iespējams, laikus jāpadomā par maskēšanas paņēmieniem. Visvienkāršākais un radikāls veids tādā gadījumā – izmantot papildus krāsas tādu sarežģītu krāsu drukai. Bet, ja budžets neļauj palielināt drukas izmaksas, mums jābūt gataviem nopietnai maketa modifikācijai, mūsu “neiejaukšanās principa” pārkāpšanai.

Trapping paņēmieni tādā režīmā ir diezgan daudzveidīgi un prasa radošu pieeju. Piemēram, gaiši violetam tekstam **C60M40** uz balta fona vēlams uztaisīt gaiši zilu kontūru. Bet smalkai salātu krāsas līnijai **C40Y95** labākais maskēšanas variants ir dzeltena kontūra. Kopumā – divkomponentu krāsām es rekomendēju ievilkt mazāku komponentu zem lielāka. Un orientēties ne tikai pēc komponentu gaišuma, bet arī pēc krāsu nobīdes minimizācijas. Lielākā trīs un četru komponentu krāsu daļa var būt optimizēta no *trapping* viedokļa. Brūno krāsu var iestatīt caur **C55M65Y65**, bet var arī caur **M30Y30K60**. Otro recepti ir krietni vieglāk maskēt. Reizēm maskēšanas elementu var pārvērst par dizaina elementu. Atkārtosos – *dry trapping* atribūtam universālas pieejas nevar būt. Rēķināties ar automātiku tādā situācijā nedrīkst. Tāds *trapping* praktiski vienmēr ir atkarīgs no attēla un tiek veikts ar roku.

Bet ir viens izplatīts gadījums, kad *dry trapping* var uzticēt automātam. Tā ir saliktā melnā krāsa (**rich black**). Ļoti bieži poligrāfijā melnās krāsas blīvuma palielināšanai, bāzes krāsai *process black* pievieno triādas sastāvdaļas. Un izmanto divu, trīs un pat četru komponentu melnās krāsas receptes¹⁰. Tādu radikālu melno krāsu bieži sauc par *superblack*, *fat black*, *rich black* (“supermelnā”, “treknā”, “bagātā”). Nesakritības kļūmes tādā melnā krāsā ir ļoti bīstamas, jo šajā gadījumā mēs vērojam vissliktāko, no artefaktu parādīšanās viedokļa, gadījumu – vislielāko kontrasta diapazonu, no baltās krāsas līdz radikāli melnai. Piemēram, baltam uz radikāli melnā fona izvietotam inversajam tekstam parādīsies krāsainās parazitējošās kontūras, kuras krasi pasliktinās skatu. Bet tieši šeit mēs varam trūkumus pārvērst labumā. Izmantojot melnās krāsas īpašību slēpt krāsu kļūmes, vienkārši ievilksim krāsainās sastāvdaļas zem melnās (15. zīm.). Tādu paņēmieni sauc par **choke rich black**.

¹⁰ Daudzās tipogrāfijās pastāv *rich black* receptūras standarti. Tajos tiek ievēroti gan kopējās krāsas daudzuma ierobežojumi (*total ink limit*) drukas laikā, gan krāsu uzklāšanas īpatnības sarežģītajās melnās krāsas bilansēs.

Pieredzējušie dizaineri jau sen izmanto ievilkšanu zem melnās krāsas. Jau pieminētajā piemērā ar inverso tekstu bija vērts vienkārši piešķirt baltajam tekstam **K100** kontūru un visi nesakritības defekti būtu nomaskēti. Analogiski var rīkoties arī citās situācijās, kad ir nepieciešama ievilkšana zem melnās. Protams, tādā gadījumā kontūrai jāpiešķir *knockout* atribūts – vēl viens iemesls baidīties no “slinkās” pogas *all black overprint*. Šādam paņēmienam ir arī trūkums, jo tipiska kontūra daudzās izdevniecības programmās veidojas no objekta centra (*center outline*). Tas izsauks ģeometrisku kropļojumu, baltais objekts kļūs nedaudz mazāks, mūsu teksts no piemēra kļūs smalkāks¹¹. Tāpēc es rekomendēju vienmēr izmantot specializēto *trapping* sistēmu iespējas, kurās ievilkšana zem melnās tiek veikta automātiski un nav sastopamas problēmas ar ģeometriju un “slinko” pogu. Šī funkcija ir tik būtiska, ka bieži vien tā darbojas piespiedu kārtā un daudzās sistēmās to nav iespējams atslēgt.

¹¹ Programmu daļā šī problēma ir risināma, kontūrai var piešķirt *outside (inside) outline*.

Tātad, *trapping* sistēmai jānāk atpazīt situāciju ar *rich black* maketā:

Objektiem ar *rich black* tiek veikta visu komponentu ievilkšana (*choke*) zem melnās.

Par supermelno tiek uzskatīta krāsa, kuras sastāvā ir melna krāsa un vēl, kā minimums, viens komponents. Atliek precizēt, ko mēs uzskatām par “melnu krāsu”. Tā nav tikai **K100**. Pareizāk būtu rēķināties ar to, ka melnās krāsas maskējošās iespējas izpaužas pat tad, kad tā ir rastrā. Es rekomendēju taisīt ievilkšanu zem melnās no 85% līmeņa (precīzāk, $BIL * 0,85$)¹². Daļa jauktu krāsu (*spot*) arī ir līdzīga melnai, piemēram, *Pantone Black*. Tāpēc “melnajā” kategorijā krāsas iekļauj pēc diviem dažādiem raksturojumiem, kurus parasti sauc par ***black trapping rules*** (*InDesign* programmā tas ir *black trap thresholds*). Vai **K** procentuālais saturs objektā ir lielāks par ***black color limit***, vai arī *ND* kādas *spot color* krāsas ir lielāks par ***black density limit***.

¹² Nevajag aizmirst par to, ka maksimālais sasniedzamais melnās krāsas līmenis drukas laikā tiek noteikts ar tehnoloģisko robežu *black ink limit* (*BIL*) un ir atkarīgs no drukas tehnoloģijas. Piemēram, avižu drukā tas parasti ir 85% līmenī.

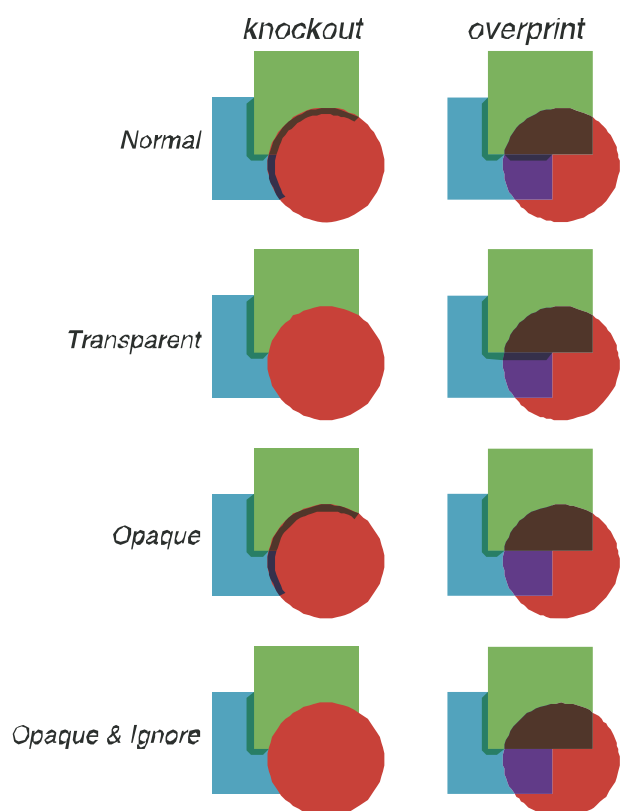
Sedzošās krāsas, lakas un citi īpašie gadījumi

Iepriekš izskatītā komponenta “spēka” analīzes metode pēc *ND* nav piemērota sedzošām krāsām. Kā jau tika izskatīts (sk. 3. zīm.), noteicošā sedzošām krāsām ir krāsu uzklāšanas secība. Gaiša krāsa, kas tiek uzklāta pēdējā, būs “stiprāka” par iepriekš uzklātu tumšo krāsu. Piemēram, “sudrabam” ir zems *ND*, bet ja šī krāsa tiek drukāta pēdējā, tad visām citām krāsām jābūt ievilkām (*choke*) zem tās. Ja drukas laikā tiek pielietotas vairākas sedzošās krāsas, korekta *trapping* veidošanai vajag pareizi noteikt to uzklāšanas kārtību – ***trap sequence*** (vai *print order*). Diemžēl tādos gadījumos ir viens stingrs ierobežojums – tikai neliela *trapping* sistēmu daļa pareizi ievēro normālu un sedzošu krāsu mijiedarbību. Pieņemts uzskatīt, ka sedzošās krāsas vienmēr ir stiprākas par parastām. Gadījumā, ja sedzoša krāsa tiek drukāta pirms normālas, tāda pieceja var izsaukt nepatīkamas kļūdas. Sistēmu daļā (piemēram, *Delta Trapper*) tādām situācijām ir paredzētas speciālas izņēmumu tabulas ***trap pair***. Pārējos gadījumos *trapping* jāveido ar roku, vai arī “jāpieregulē” *ND* tādā veidā, lai piemānītu automātu.

Bez darba specifikas ar sedzošām krāsām ir arī daudz citu variantu, kuros ir nepieciešama īpaša *trapping* veidošanas vadīšana. Piemēram, *overprint* var būt pielietots gan sedzošiem objektiem, gan caurspīdīgām lakām (formas fragmentārai lakošanai). Abos gadījumos pieceja *trapping* atribūtam atšķiras. Šiem nolūkiem ieviesīsim speciālu krāsas atribūtu – *opacity*. Tam ir četri apzīmējumi:

- ***normal*** – parasta ofseta puscaurspīdīga krāsa, *trapping* veidošana pēc parastiem noteikumiem;
- ***transparent*** – caurspīdīga laka;
- ***opaque*** – sedzoša krāsa;
- ***opaque & ignore*** – īpašs sedzošās krāsas pielietošanas gadījums (sk. turpmāk).

16. zīmējumā parādīts parasto *knockout* un *overprint* atribūtu un mūsu specifiskā krāsas atribūta mijiedarbības piemērs. Paņemsim divus kvadrātus – gaiši zilu un gaiši zaļu. Tie ir nokrāsoti ar parastām krāsām. Gaiši zilais kvadrāts ir tumšāks par gaiši zaļo, tāpēc gaiši zaļais tiek ievilkts zem gaiši zilā. Pēc tam diviem kvadrātiem tiek pievienots specifisks objekts – sarkans aplis. Labākai pārskatāmībai tas uztaisīts caurspīdīgs, bet tā var būt metalizētā krāsa, spieduma klišeja, fragmentārās lakošanas maska. Trešajam objektam var būt *overprint* atribūts. Bet galvenais – šis objekts var iespaidot pārējo divu *trapping*. Lai vadītu *trapping*, piešķirsim trešajam objektam speciālu *opacity* atribūtu. Iegūstam astoņus (precīzāk – sešus) variantus:



16. zīm. *Opacity* atribūts

1. Sarkanajam aplim ir piešķirti *normal* un *knockout* atribūti. Veidojas parastais *trapping* pēc *ND* noteikumiem. Kā redzam, sarkanais aplis ir vistumšākais objekts (ar visaugstāku *ND*), tāpēc abi kvadrāti tiek ievilkti zem tā.

2. *Normal* un *overprint* atribūti. Šajā gadījumā *trapping* aplim neveidojas (tas vienkārši nav iespējams – iedobuma zem apla nav). Nedaudz ir mainījies kvadrātu pāra *trapping*. Punktā, kur saiet visi trīs objekti, izveidojusies sprauga. Tāda efekta būtība tiks izskaidrota vēlāk, *trap geometry* sadaļā.

3. *Transparent* un *knockout* atribūti. Īpašs gadījums un pēc pirmā acu uzmetiena – savāds. It kā absolūti caurspīdīgai lakai ir nepieciešams *overprint*, bet poligrāfijā iespējamas vēl divvainākas lietas. Šo variantu izmanto, ja ir nepieciešams aizliegt *trapping* objektu daļai.

4. *Transparent* un *overprint* atribūti. Tas ir fragmentārās lakošanas variants. Aplim *trapping* netiek veidots. Aplis neiespaido kvadrātu *trapping*.

5. *Opaque* un *knockout* atribūti. Šajā gadījumā specifiskajam objektam *trapping* veidojas pēc noteikuma “sedzošā krāsa vienmēr ir stiprāka par parasto”.

6. *Opaque* un *overprint* atribūti. Ir līdzīgs 4. variantam, bet šajā gadījumā mūsu specifiskais objekts iespaido kaimiņus. Zem sarkanā apla ir pazudis kvadrātu pāra *trapping*. Loģika ir vienkārša – ja objekts ir sedzošs, nesakritības problēmas zem tā nav aktuālas, jo caur to neko nevarēs redzēt. Kādēļ tad tur veidot *trapping*.

Brīžiem rodas situācija, kad mūsu specifiskajā krāsā nokrāsotiem objektiem pilnībā ir jāliedz veidot ar sevi *trapping*. Tāds gadījums ir raksturīgs objektiem, kuriem *trapping* tika veidots ar roku. Tieši tam tiek izmantots specifisks *opaque & ignore* atribūts.

3. daļa. *Trapping* veidošana

Mēs iemācījāmies pieņemt lēmumu par *trapping* vajadzību. Tagad noskaidrosim – kādu tieši. Atgādināšu, ka pārsvarā aprakstu darbu krāsu pāra (*wet trap*) režīmā.

1. *Elementa platums trap width*

Šo platumu nosaka mūsu gaidāmā nesakritība. Es jau atzīmēju, ka par normālu drukas laikā tiek uzskatīta nesakritība rastra puslīnijas platumā. Daļā tipogrāfiju pieeja var atšķirties, tāpēc tieši saistīt *trap width* ar drukas liniatūru nebūtu korekti. Kopumā var teikt, ka augstas klases produkcijai tipiska ir nesakritība ne vairāk kā par 80-90 mkm. Vai aptuveni 0,25 pt – krusta biezums sakritības zīmes (arī griešanas) vietā. Protams, šāda pieeja ir vienkāršota. Piemēram, uz planetārās divkrāsu iespaidmašīnas var gaidīt diezgan labu sakritību starp pirmā un otrā laidiena krāsu pāriem. Un, pamatā, vēltīt uzmanību sakritībai starp laidieniem. Tādu pieeju sauc par “trepju” *trapping*. Fleksogrāfijas drukas veida gadījumā lielas problēmas krāsu sakritībā izsauc elastīgo iespaidplašu ģeometriskie kropļojumi, tāpēc gaidāmās nesakritības materiāla platumā un garumā var būt dažādas.

Gaidāmās nesakritības lieluma vērtējums ir atkarīgs no daudziem faktoriem. Krāsainajā drukā nesakritību ietekmē gan drukas procesa ģeometriskie kropļojumi, kas var atšķirties katrai sekcijai (krāsai), gan nesakritības problēmas starp sekcijām (reģistrs) un starp laidieniem. Sniegt universālas rekomendācijas šajā gadījumā ir ļoti grūti. Var tikai atzīmēt, ka lieltirāžas produkcijas (piemēram,

makets

pre
press



auto amount +

pre
press

17. zīm. *Trapping*
QuarkXpress programmā

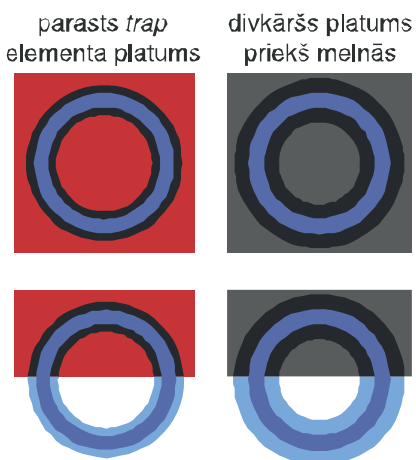
etiķešu/iepakojuma) ražošanai *trapping* tiek veidots rūpīgāk un bieži tiek ņemts vērā vairākums ar sakritību saistīto problēmu. Žurnālu un reklāmas produkcijas sagatavošanas laikā biežāk tiek pielietoti nedaudz vienkāršoti paņēmieni, tai skaitā “tipiskie”, ar nedaudz pazemināto *trap width*¹³.

¹³ Gaidāmās nesakritības aprēķinos tiek pielietotas statistikas metodes.

Vidējā (vai tipiskā) nesakritības lieluma vērtība pieļauj, ka tirāžā var būt sastopamas tā saucamās “pierīkošanas” loksnes, kurās ir gadījusies lielāka nesakritība, nekā bija gaidīts. Ne katram produkcijas veidam ir pieļaujama tādu “pierīkošanas” lokšņu parādīšanās tirāžā, tāpēc etiķešu/iepakojuma (piemēram) ražošanā parasti izmanto lielākas *trap width* vērtības.

Īpaši gribētos atzīmēt, ka šāds paņēmieni var būt pielietots tikai adekvātā *trapping* algoritma izmantošanas gadījumos, kad *trap* elements rodas tikai uz krāsu pāra robežas. Tikai neliela programmu daļa ir spējīga veidot tādu *trapping*. Piemēram, nevajag izmantot šīs rekomendācijas tādām programmām kā *QuarkXpress*. Piešķirot *trapping* atribūtu ar *trapping QuarkXpress* (kā arī daļā citu programmu), tas

mainīs izmēru visam objektam (!) par *trap width*¹⁴ vērtības lielumu. Kas var nozīmīgi mainīt maketa izskatu. Piemēram, *trapping* veidošanas laikā par teksta elementu *QuarkXpress* programmā tiek uzskatīta rindiņa. Ne vairāk, ne mazāk – tieši rindiņa. Atcerēsimies 9. zīmējumu. Pielietojot normāla 10 pt teksta 0,25 pt lieluma resnājumu, mēs iegūsim pustrekno drukas stilu (*bold*). Mums jāatceras, ka, neskatoties uz visām viltībām, uztaisīt *keyline* pilnīgi nemanāmu nav iespējams. Palielināt biezumu “ar rezervi” uzstājīgi nerekomendēju. Bet ar vienu izņēmumu.



18. zīm. Platuma noteikumi melnai krāsai

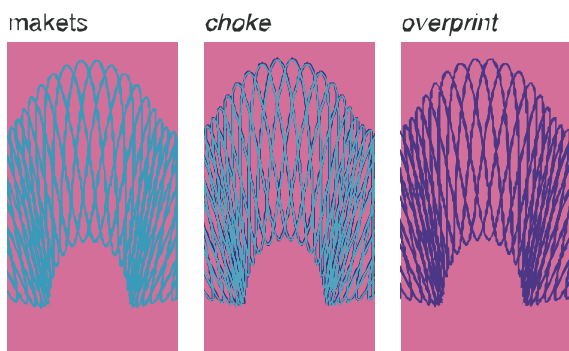
bez kontroles



small edge control



19. zīm. Smalku elementu kontrole



20. zīm. Kropļojums smalkajās līnijās

uz visām viltībām, uztaisīt *keyline* pilnīgi nemanāmu nav iespējams. Palielināt biezumu “ar rezervi” uzstājīgi nerekomendēju. Bet ar vienu izņēmumu.

¹⁴ *QuarkXpress* programmā pēc noklusēšanas tiek izmantots 0.144 (50 mkm) *trap width* platums. Neskatoties uz to, ka tā ir klāji pazemināta vērtība, mainīt to es neiesaku.

2. Atsevišķs noteikums melnajai krāsai *black trap width*

Melnai krāsai “rezerve” netraucēs. Mēs taču atceramies, ka krāsas kropļojumi zem melnās nav tik pamanāmi, kā pārējās krāsās. Tāpēc mēs varam bez īpaša riska palielināt *trap* objektu (*keyline*) platumu, ievelkot tos zem melnās, pat 1,5-2 reizes. ***Black width scaling*** parametrs ļauj vadīt papildu rezervi platumā melnajai krāsai. 18. zīmējumā ir parādīts, ka zils riņķis paliek platāks uz sarkana un uz melna fona, pielietojot *trapping*.

3. Sīko elementu kontrole *small edge control*

Iespējamās situācijas, kad objektu izmēri ir salīdzināmi ar *trap width*. Lai izvairītos no kļūdām, kas rodas nesakritības gadījumā (19. zīm.), tādās vietās *trapping* platumam jābūt proporcionāli samazinātam. Sistēmās, kas ir uzbūvētas uz *Adobe InRip Trapping* bāzes, visi sīkie elementi tiek pārbaudīti pēc noteikuma:

Trap width nevar pārsniegt pusi no sīkā elementa platumā.

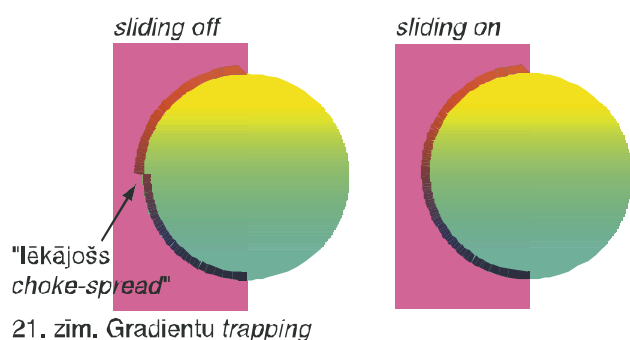
Dažkārt sīkiem objektiem labāk izmantot *overprint*. Būtībā, *keyline*, kas rodas *trapping* iestatīšanas rezultātā, ne tikai ievieš zināmas krāsu nobīdes objektā, bet arī iespaido kopējo attēla izskatu. Kamēr objektam ir liels (salīdzinoši ar *keyline*) izmērs, šī iespaids iespēju var neievērot. Bet, kad objekta izmēri kļūst salīdzināmi ar *keyline*, tas var izsaukt būtiskas kļūdas (20. zīm.). Šādos gadījumos mēs esam spiesti kaut ko ziedot – vai nu pieļaut krāsu kropļojumus sīku objektu *overprint* vietās, vai nu samierināties ar zīmējuma vizuālām deformācijām. Ņemot vērā to, ka deformācijas šādos gadījumos ir neizbēgamas, risinājumus vēlam pieņemt vēl ilustrāciju dizaina veidošanas laikā. *Heidelberg Supertrap* šādi tiek kontrolēts *choke* priekš smalkām līnijām. Pēc *overprint black* līdzīga noteikuma (sk. 1.daļu) – *stroke*, kuru platums nav lielāks par noteikto (caur ***line split*** parametru), *trapping* tiek nomainīts uz *overprint*. Citu sistēmu lielākajā daļā šāda kontrole jāveic ar roku.

4. *Trapping virziens*

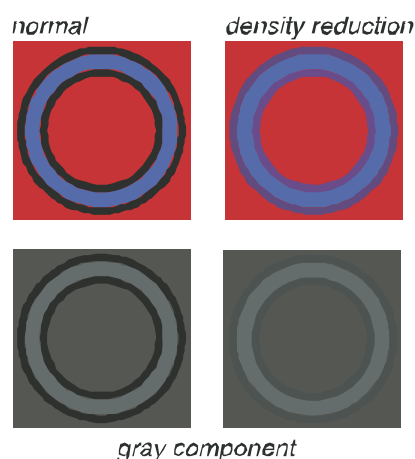
Atgādināšu *contour-defining* noteikumu – objekta vizuālo formu nosaka tā kontūra. Kontūra veidojas divu elementu – objekta un tā projekcijas – savietojuma vietā. Vienkāršākā gadījumā objekts un projekcija ir stingri vienādi pēc izmēra – tas ir *knockout*. Ja projekcija ir mazāka par objektu – tas ir *choke*. Ja objekts paliek lielāks, bet projekcija nemainīga – tas ir *spread*. Divos pēdējos variantos veidojas krāsu pārklāšanās – *keyline*. Tieši *keyline* arī nosaka mūsu kontūru, tas ir, nosaka objekta vizuālos izmērus. Tāpēc ir svarīgi, lai *keyline* saglabātu dizaina

izstrādes laikā noteiktos objektu izmērus. Ja objektu pāri augšējais ir tumšāks par apakšējo, tad augšējais objekts nosaka izmērus, bet apakšējais būs ievilkts (*choke*) zem tā. Ja otrādi – objektu pāri augšējais

objekts ir gaišāks par apakšējo, tad izmērus noteiks apakšējais, bet augšējais būs paplašināts (*spread*). Kurš no objektiem ir gaišāks/tumšāks, nosaka to *ND* (parastām krāsām).



21. zīm. Gradientu trapping



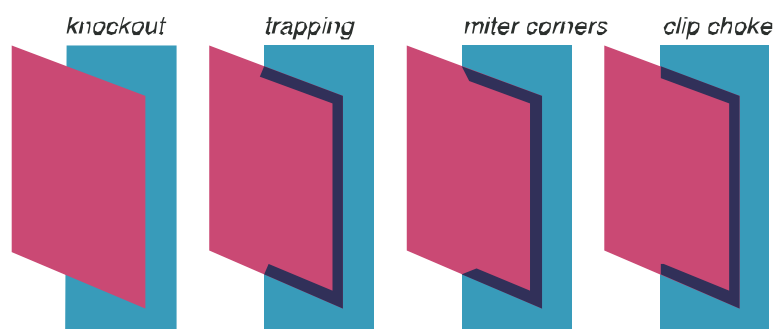
22. zīm. Keyline redzamības samazināšana

	Color 1	Color 2	Normal	ND Reduction 100%
C	0 %	100 %	100 %	77 %
M	100 %	60 %	100 %	78 %
Y	100 %	0 %	100 %	70 %
K	0 %	0 %	0 %	0 %
ND	0.66	0.82	1.17	0.82

Krāsas daudzuma samazināšanas pakāpe ir regulējama. Pati modernākā metode – *trap color density reduction*. Tā samazina *keyline ND* līdz tumšāka pāra objekta līmenim, tādā veidā pilnībā izslēdzot košuma lēcienus *trapping* joslā.

6. Trap geometry vai kādas formas jābūt *keyline*

Šo diezgan sarežģīto jautājumu sadalīsim trīs vienkāršākos: kādas formas nobeigumiem jābūt *keyline*, kādā formā stūriem, un kā rīkoties divu elementu *trapping* savienošanas vietās. Vienkāršākā gadījumā



23. zīm. Keyline nobeiguma forma

optimālā nobeiguma forma (vismazāk pamanāma) ir 45°, tā saucamā *miter*. Sarežģītākajos gadījumos (23. zīm.) reizēm labāk izmantot to formu, kuru nosaka objekts, tā saucamo *clip choke*. Bet, pielietojot *clip choke*, ir jāievēro piesardzība. Šis paņēmieni bieži izsauc kļūdas (24. zīm.). Uz *Adobe InRip* bāzētās sistēmās tiek izmantots elegants nobeiguma formas vadības algoritms, kas

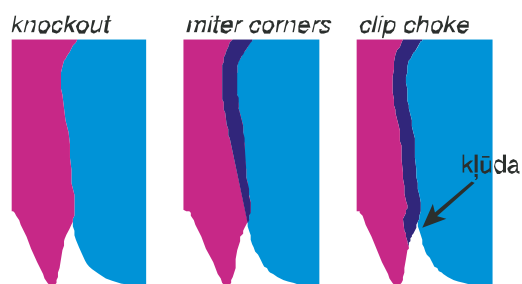
Bet, ja objekti pāri ir vienādi vai ļoti tuvi pēc *ND*? Tad *trapping* labāk veidot simetriski uz abām pusēm no objektu malām – tas ir *centerline* režīms.

Centerline trap limit parametrs nosaka, cik vienādiem pēc *ND* jābūt objektiem pāri. 100% (ideāli vienādi) sakritības gadījumā, *centerline* veidosies ļoti reti. 0% sakritības gadījumā visi *trap* objekti (izņemot *rich black*) tiks veidoti *centerline* režīmā. Pēc manām domām, vislabākā ir 90% vērtība. Tādas vērtības gadījumā *centerline trap limit* samazinās “lūkājošā *choke-spread*” (21. zīm.) artefakta parādīšanās

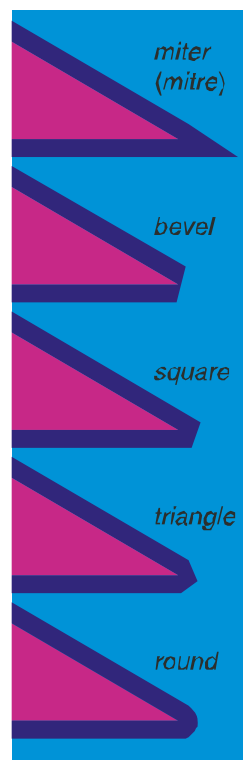
iespēju. Šis artefakts mēdz rasties, piešķirot *trapping* atribūtu gradientam. Attīstītās *trapping* sistēmās “lūkājošā *choke-spread*” artefaktu kontrolē speciāls algoritms, kura jutīgumu vada *slidilg traps color limit* (*shift limit Trapwise* ptogrammā) parametrs.

5. Kādā krāsā jābūt *keyline*

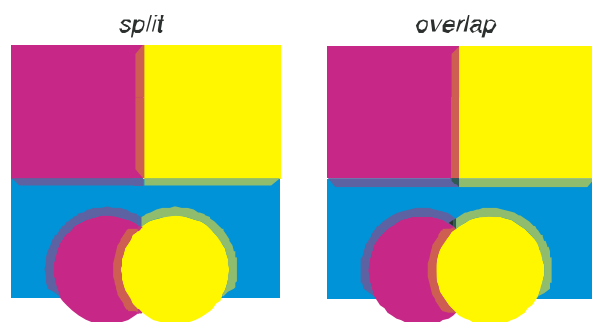
Vienkāršākais variants – *keyline* krāsu nosaka maksimālās pāra komponentu vērtības. Bet šajā gadījumā *keyline* krāsa būs pārāk tumša (22. zīm.). Tas izjauc mūsu “svēto” neiejaukšanās principu. Tāpēc bieži tiek pielietotas modernākas *keyline* krāsas aprēķināšanas metodes, tas samazina tās pamanāmību. Dažādās sistēmās tām ir atšķirīgi nosaukumi: *tint reduction*, *trap color reduction*, *trap color scaling*.



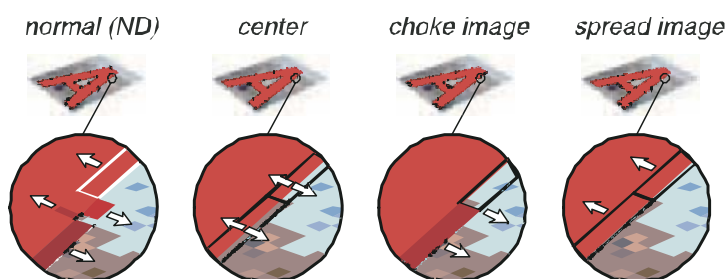
24. zīm. Kļūda clip choke gadījumā



25. zīm. Stūra forma



26. zīm. Uzklāšanas vadība



27. zīm. Trapping virziens darbā ar rastriem

apvieno *miter* un *clip choke* priekšrocības un neizsauc šādas kļūdas.

Stūru formu (25. zīm.) sarežģītos *keyline* gadījumos vada *trap corner shape* (*line join Supertrap* programmā, *trap appearance join style InDesign* programmā). Kādu tieši formu izvēlēties? Protams – minimāli pamanāmu. Vienkāršākā *miter* metode var izsaukt pietiekami pamanāmas kļūmes, īpaši mazāku par 30° leņķu gadījumā. Nogriezt stūri ar *bevel* metodi – arī nav labākais risinājums, tā slikti nomaskē nesakritības. Tāpēc šeit ir populārākas modernākās aso stūru nolīdzināšanas

metodes, tādas kā *square*, *triangle* un *round*.

Līdz šim mēs izskatījām vienkāršāku gadījumu – pāra *trapping*. Bet kas notiek gadījumā, ja objektu ir vairāk nekā divi? Piemēram, trīs (26. zīm.). Viegli iedomāties, ka tādā gadījumā iespējamās situācijas, kad *trapping* elementi uzklāsies viens otram. Šādas uzklāšanas (*overlap*) gadījumā ir iespējama nevēlamo tumšo punktu parādīšanās. Bet, ja mēs izvairīsimies no šādu punktu parādīšanās (*split*), kļūs iespējama baltu punktu parādīšanās. Tāpēc daļā *trapping* sistēmu šī situācija tiek speciāli vadīta. *InDesign* programmā tas ir *trap appearance end style*. Ja kaut vai viens no objektiem ir pietiekami tumšs, labāk izmantot *overlap*. Tumšs punkts vizuāli pievienosies tumšajam objektam un defekts praktiski nebūs pamanāms. Bet ja visas krāsas ir košas vai gaišas – labāk izvairīties no pārklāšanās.

7. Rastra attēlu trapping

Līdz šim mēs izskatījām darbu ar vektora un teksta objektiem. Rastra attēlu *trapping* ir uz to pašu principu pamata, bet tam arī ir savas īpatnības. Izveidot augstas kvalitātes *trapping*, izmantojot rastra attēlus, ir ļoti sarežģīti. Izstrādājot dizainus dažādām etiķetēm, iepakojumam un citai produkcijai ar augstām prasībām pret krāsu sakritību un *trapping*, es rekomendēju maksimāli izmantot vektora attēlu formātus. Un parastai komercprodukcijai vēlams visus ieskanētos logotipus, klišejas un citus objektus pārveidot vektoros. Bet gadās situācijas, kad ir nepieciešams *trapping* ar rastra attēla izmantošanu. Pastāv trīs rastra attēlu *trapping* varianti:

- pāra objekts (vektors) – attēls (rastrs) *trapping*;
Atcerēsimies piemēru sadaļā “Pielietojuma prakse” (6. zīm.). Mūsu vektora objekts – tā ir liela klišeja, kurai ir bīstami piešķirt *overprint* atribūtu. Spiedums iet pāri attēlam. Spieduma pierīkošanas precizitāte nav augsta, tāpēc mums jākompensē attēla ievilkšana zem klišejas par pāris milimetriem. Otrais piemērs – fona lomu pilda metalizētā krāsa. Pilnkrāsu attēliem uz tāda fona arī ir nepieciešams *trapping*. Līdzīgos gadījumos tas tiek ieslēgts ar ***trap object to images*** opciju.

- pāra attēls (rastrs) – attēls (rastrs) *trapping*;

Bet, ja mūsu piemērā klišeja nebūtu vektorizēta un tiktu ielikta maketā rastra veidā (*contone*). Vai steigā aizmirsa vektorizēt logotipu un tas atrodas virs rastrēta attēla. Parastajā gadījumā – ja liels, tipiski “vektora” dizaina elements kaut kādu iemeslu dēļ ir uztaisīts rastrā un pārklājas ar citu rastra attēlu – ieslēdzam opciju ***trap images to images***.

- *trapping* attēlā.

Ir ļoti nevēlams izmantot šo metodi. Normālam rastrētām attēlam rezultāts būs briesmīgs – pazaudēts asums, pazaudēts kontrasts, citi artefakti. Priekš kam tad daudzās programmās ir opcija **trap within image**? Īpašiem gadījumiem. Piemēram, kad visa maketēšana ir uztaisīta rastra redaktorā (*PhotoShop*) un saglabāts šis brīnummakets TIF formātā.

Trapping virzienu darbā ar rastriem bieži vien nākas iestatīt ar roku. Tādos gadījumos attēla gaišums *trapping* zonā var krasi mainīties pat atsevišķu pikseļu līmenī (25. zīm.). Un parastā virziena izvēles metode pēc *ND* izsauks nepieņemamus *spreads-choke* lēcienus. Es rekomendēju izmantot *center* metodi. Gan gadījumā, kad abu objektu *ND* uz robežas ir tuvu, gan gadījumā, kad tie regulāri un krasi mainās. *Choke* un *spreads* metodes var izmantot tikai tad, kad viena objekta gaišums ir neapšaubāmi augstāks (zemāks) par otrā objekta gaišumu visas robežas garumā, vai arī viens no objektiem ir sedzošs.

Gadījumu daļā (piemēram, izmantojot *copydot*) rastrētais attēls var būt piedāvāts melnbaltā izpildījumā (*one bit*). Šāda formāta apstrāde tiek ieslēgta ar atsevišķu opciju **trap one bit images**.

Paraksti pie attēliem:

1. zīm. Nesakritība pie drukas
2. zīm. Divi drukas varianti / iedobums – *knockout* / uzklāšana – *overprint*
3. zīm. Krāsu nobīdes uzklāšanas laikā
4. zīm. Melnā krāsa – *knockout* un *overprint*
5. zīm. Lielas burta bieznes gadījumā *overprint* kļūs redzams
6. zīm. Sedzošā baltā krāsa uz caurspīdīgas pamatnes / baltā *overprint* / baltā *knockout*
7. zīm. *Black overprint Agfa Apogee* procesorā
8. zīm. *Heidelberg Supertrap 4.0*.
9. zīm. Kļūda “teksts pa tekstu” *QuarkXpress* programmā / makets / rezultāts “pēc noklusēšanas” / izlabotais rezultāts
10. zīm. Krāsu lamatas / *knockout* / *trapping* / ideāla sakritība / nesakritība
11. zīm. Vērtējums pēc pelēkā komponenta
12. zīm. Divi *trapping* virzieni / *spread* / *choke* / dzeltenā krāsa paplašinās attiecībā pret savu projekciju tumšajā fonā / projekcija gaišajā fonā ir mazāka par zilo pamatobjektu
13. zīm. Krāsu pāra kopējais komponents / color 1 / color 2 / *common density* / CMYK / CMYK / CMYK
14. zīm. Krāsu artefakti / *knockout* / *trapping* / ideāla sakritība / nesakritība
15. zīm. Ievilkšana zem melnās / *knockout* / *choke rich black* / ideāla sakritība / nesakritība
16. zīm. *Opacity* atribūts / *knockout* / *overprint* / *Normal* / *Transparent* / *Opaque* / *Opaque & Ignore*
17. zīm. *Trapping QuarkXpress* programmā / makets / *auto amount* +
18. zīm. Platuma noteikumi melnai krāsai / parasts *trap* elementa platums / divkārtšs platums priekš melnās
19. zīm. Smalku elementu kontrole / bez kontroles / *small edge control*
20. zīm. Kropļojums smalkajās līnijās / makets / *choke* / *overprint*
21. zīm. Gradientu *trapping* / *sliding off* / *sliding on* / “lūkājošs *choke-spread*”
22. zīm. *Keyline* redzamības samazināšana / *normal* / *density reduction* / *gray component*
23. zīm. *Keyline* nobeiguma forma / *knockout* / *trapping* / *miter corners* / *clip choke*
24. zīm. Kļūda *clip choke* gadījumā / *knockout* / *miter corners* / *clip choke* / kļūda
25. zīm. Stūra forma / *miter (mitre)* / *bevel* / *square* / *triangle* / *round*
26. zīm. Uzklāšanas vadība / *split* / *overlap*
27. zīm. *Trapping* virziens darbā ar rastriem / *normal (ND)* / *center* / *choke image* / *spread image*