

Mit den Jahren wird man im Allgemeinen schlauer und so stellt meine Diplomarbeit nur den Stand meines damaligen Wissens dar. Inzwischen habe ich keine fundamentalen Veröffentlichungen zum Thema mehr gefunden. Sollte jemand etwas wissen, bin ich für Hinweise dankbar.

Als wichtige Ergänzung möchte ich auf einen Umstand hinweisen, auf den ich erst ein paar Jahre nach meiner Diplomierung gestoßen bin: Jedes Raster enthält auch alle ganzzahligen Vielfachen als Raster. Ein Raster von 60 Linien/cm enthält beispielsweise auch die Raster von 30 Linien/cm und 15 Linien/cm. Diese Raster müssen bei Moireberechnungen mit berücksichtigt werden, wohingegen die in der Diplomarbeit erwähnten sekundären Moires wahrscheinlich nur eine untergeordnete Rolle spielen.

Am besten dazu noch meine Veröffentlichungen im Deutschen Drucker lesen, die noch etwas über die Diplomarbeit hinausgehen.

DD 21-22/92, w70; Vierfarbenraasterung - die Sache mit Rosetten, Farbdrift und Moiré, Teil 1;

DD 43/92, w12; Probleme bei Vierfarbenraasterung: Rosetten, Farbdrift und Moiré, Teil 2

DD 35/93, w44 Vierfarbenraasterung - die Sache mit Rosetten, Farbdrift und Moiré, Teil 3

Die meisten Patente zur Raasterung waren bereits zur Zeit der Diplomierung schon ein paar Jahre alt, so dass sie bereits oder demnächst auslaufen dürften. Wobei ich gespannt bin, welche verbesserten Raster uns noch erwarten.

Mittweida, 5. September 2005

Christian Greim

Diplomarbeit im Studiengang Druckereitechnik

Titel:

Probleme der Vierfarbrasterung  
Mathematische Grundlagen  
bis zur  
digitalen Lösung

Vorgelegt von:

Christian Greim

Hiermit versichere ich, die vorliegende Diplomarbeit ohne fremde Hilfe angefertigt zu haben. Alle benutzten Quellen sind im Literaturverzeichnis oder direkt im Text vollständig aufgeführt.

Stuttgart, 3. Januar 1992

---

(Christian Greim)

---

|  |    |
|--|----|
| Vorwort .....  | 1  |
| 1. Probleme im Detail .....  | 2  |
| 1.1. Begrenzung des Tonwertumfangs durch die Rasterfeinheit .....                          | 2  |
| 1.2. Ungewollte Erscheinungen im Übereinanderdruck .....                                   | 3  |
| 1.3. Bisherige Lösungsvorschläge .....   | 5  |
| 1.3.1. OPTI-Rastersystem .....   | 5  |
| 1.3.2. Frequenzmodulierter Raster .....  | 6  |
| 1.3.3. Sonstige .....  | 8  |
| 2. Genaue Problemanalyse .....   | 10 |
| 2.1. Bisheriger Stand .....  | 10 |
| 2.2. Zusammenhang zwischen Kontrast der Druckfarbe mit dem Hintergrund und Auflösung ..... | 11 |
| 2.2.1. Grundsätzliches .....   | 11 |
| 2.2.2. Experiment .....  | 12 |
| 2.2. Analyse von Rosette, Moire und Farbdrift .....  | 14 |
| 2.2.1. Einführung .....  | 14 |
| 2.2.2. Überlegungen zur Formel nach Tollenaar .....  | 15 |
| 2.2.2.1. Formel nach Tollenaar.....  | 15 |
| 2.2.2.2. Folgerungen aus der Formel .....  | 16 |
| 2.2.2.3. Verbesserte Anwendung der Formel .....  | 18 |
| 2.2.2.4. Beschreibung von Rastern im Hinblick auf die Formel .....                         | 20 |
| Kurvenschar Moireweite nach Punkt 2.2.2.3. ....  | 22 |
| Kurvenschar Winkelabweichung nach Punkt 2.2.2.3. ....                                      | 23 |
| 3. Neue Rastermöglichkeiten anhand der Geometrie der Tiefdruckgravur .....                 | 25 |
| 3.1. Betrachtung des Tiefdruckrasters .....  | 25 |
| 3.2. Theoretische Optimierung .....  | 26 |
| 3.3. Computerprogramm zur Berechnung von Moires bei gravurähnlicher Anordnung.....         | 27 |
| 3.4. Anwendung des Ergebnisses und des Programms .....                                     | 31 |
| 4. Überlegungen zur Gestaltung neuer Raster.....   | 34 |
| 4.1. Grenzen regelmäßiger Rasterung.....   | 34 |
| 4.2. Neue Lösungen .....   | 34 |
| 4.2.1. Der Schritt von der Regelmäßigkeit zum Chaos .....                                  | 34 |
| 4.2.2. Wie chaotisch muß ein Raster sein? .....  | 35 |
| 4.2.3. Erster prinzipieller Vorschlag .....  | 36 |
| 4.2.3.1. Gedankenmodell .....  | 37 |
| 4.2.3.2. Computerprogramm zur Erzeugung unregelmäßiger Rastermo-                           |    |

---

|   |    |
|---|----|
| delle.....  | 38 |
| 4.2.3.3. Realisierung des Rastermodells .....                           | 47 |
| 4.2.4. Versuch in Postscript .....                                      | 49 |
| 4.2.4.1. Grundsätzliche Überlegungen .....                              | 49 |
| 4.2.4.2. Zur Programmiertechnik .....                                   | 50 |
| 4.2.4.3. Das Postscriptprogramm .....                                   | 53 |
| 4.2.4.4. Zum Vorprogramm .....  | 59 |
| 4.2.4.5. Das Vorprogramm .....  | 61 |
| 4.2.4.6. Weitere Vorteile des Rasters .....                             | 65 |
| Abbildung eines groben Postscriptrasters nach Punkt 4.2.4.2. ....       | 67 |
| Abbildung eines statistisch verzerrten Rasters nach Punkt 4.2.4.6. .... | 67 |
| 5. Zur Praxis des Belichtens von Rastern .....                          | 68 |
| 5.1. Arbeiten an Postscriptsystemen .....                               | 68 |
| 5.2. Auswertung der Ergebnisse .....                                    | 70 |
| Testdruck für Experiment nach Punkt 2.2.2. ....                         | 71 |
| Vierfarbdruck mit statistisch verzerrtem Raster nach Punkt 5.2. ....    | 72 |
| Literaturverzeichnis .....  | 73 |

## Vorwort

Das Problem bei dieser Diplomarbeit ergab sich aus dem Flexodruck. Besonders der Flexodruck hat das Problem, daß sehr kleine Punkte nicht mehr gedruckt werden. So ist nur eine grobe Rasterung möglich, wenn man noch genügend Zeichnung in den hellen Tönen erhalten will. Die Rasterweite liegt bei etwa 30 Linien pro Zentimeter. Bisher wurden Farbauszüge immer von sehr teuren Scannern erzeugt. Inzwischen setzt sich aber für die Ausbelichtung immer mehr der Laserbelichter durch, der im Zusammenhang mit DTP-Programmen auch den Text mit ausbelichtet. Die bisherigen Programme simulieren bei der Bildbelichtung mehr oder weniger nur die bisherigen photomechanischen Verfahren. Dabei setzt der Belichter den einzelnen Rasterpunkt aus mehreren kleineren Rasterpunkten zusammen. Diese feinsten Rasterpunkte erreichen inzwischen eine Feinheit von bis zu 1333 Linien pro Zentimeter. Aus diesen können theoretisch beliebige Strukturen erzeugt werden, um ein Bild im weitesten Sinne des Wortes zu rastern. Die Möglichkeiten, hier Neues auszuprobieren, sind noch lange nicht ausgeschöpft. Vorteile daraus kann nicht nur der Flexodruck ziehen, sondern auch der Offsetdruck, wenn er mit sehr hoher Rasterzahl arbeiten will. Die Ergebnisse dieser Diplomarbeit sollen so allgemein sein, daß sie auf alle autotypischen Druckverfahren angewandt werden können, die als Ausgangsmaterial in der Druckvorstufe einen Film benötigen.

Dazu sollen zunächst die theoretischen Grundlagen noch einmal etwas genauer betrachtet werden als das bisher der Fall war. Im elektronischen Bildbereich gibt es darüber schon gute Literatur, zum Beispiel das Buch von Friedrich Bestenreiner "Vom Punkt zum Bild", aber die elektronischen Medien waren von Beginn an wissenschaftlicher ausgerichtet als der Druckbereich, in dem eher eine gewisse Bastlermentalität herrscht. Nach diesen theoretischen Betrachtungen sollen dann einige Vorschläge für neue Rasterungsarten, oder verbesserte Versionen herkömmlicher Rasterungsarten gemacht werden. Für den Flexodruck und andere Druckverfahren müssen allerdings die optimalen Rasterweiten und die dazugehörigen Druckkennlinien erst noch ermittelt werden. Dies auch noch zu behandeln, würde den Rahmen einer Diplomarbeit bei weitem sprengen.

Diese Diplomarbeit konnte wegen der Vielschichtigkeit des Themas keine geschlossene, zielgerichtete Arbeit werden. Sie versucht im ersten Teil erst einmal bisherige Ergebnisse zusammenzufassen und neu zu bewerten. Interessant dabei ist vielleicht der Übergang vom herkömmlichen zum Tiefdruckraster

und die erstmals fundierte abweichende Verwendung der gelben Druckfarbe. Der weitere Teil spiegelt die eigenen Überlegungen wieder, wie ein neues Raster aussehen könnte, wobei auch die wichtigsten Sackgassen mit angegeben wurden. Diese traten meistens nicht wegen prinzipieller Schwierigkeiten auf, sondern wegen Schwierigkeiten in der Realisierung. Ein Beispiel sollte durchexerziert werden. Die anderen Wege können aber trotzdem eine Anregung sein, in dieser Richtung weiterzudenken. Die Möglichkeiten, die die neuen digitalen Techniken in dieser Richtung eröffnen, sind immens, so daß sich die derzeitige Suche nach der optimalen Rasterpunktform in Postscript, die zur Zeit die Gemüter bewegt, nur noch marginal ausnimmt. Deshalb konnte diese Diplomarbeit nur ein paar allgemeine Wegweiser in diesem riesigen neuen Feld setzen.

## 1. Probleme im Detail

### 1.1. Begrenzung des Tonwertumfangs durch die Rasterfeinheit

Punkte unter einer gewissen Größe drucken nicht mehr. Die Größe, ab der nicht mehr gedruckt wird, ist bei jedem Druckverfahren unterschiedlich. Sie ist aber absolut, das heißt, sie variiert nicht mit der Rasterweite. Hinzu kommt ein analoger Effekt bei den Tiefen. Ab einer bestimmten Größe der nicht druckenden Stellen ( "weiße Punkte" ) schmieren diese Punkte zu. Beim Flexodruck ist es sogar so, daß ein dichter Raster im Druck eine höhere Flächendeckung ergibt als ein glatt schwarzer Ton, weil es zu einer Art Tiefdruck kommt. Aus diesen Gründen nimmt der wirklich druckende Tonwertumfang gegenüber dem Tonwertumfang im Film bei höher auflösendem Raster ab. Je größer aber ein Raster ist, desto mehr fallen die Strukturen des Rasters auf. Das Gehirn integriert zwar aus den einzelnen Punkten Grauwerte oder Farben, nimmt aber, wenn die Auflösung nicht deutlich unter der wahrnehmbaren Schwelle der Augen liegt, zusätzlich die Rasterstrukturen wahr, was natürlich den Bildeindruck stört.

Zum Verlust des Tonwertumfangs einige Beispiele: Die optimale Übertragung von Punkten im Offsetdruck beinhaltet, daß Punkte mit einem Durchmesser von 10 µm gerade noch wiedergegeben werden diese Größe stammt aus den FOGRA-Mitteilungen Nr. 141/142, Seite 19. Die Größenangabe bezieht sich nur auf die Filmkopie. Im Druck ist mit einer ziemlichen Tonwertzunahme zu rech-

nen. Die Punkte haben also eine Fläche von  $(5 \text{ um})^2 \times \pi$ , also  $79 \text{ um}^2$ . Eine Rasterzelle im 60er Raster hat eine Fläche von  $(\frac{1}{60} \text{ cm})^2$ , also insgesamt  $27\,556 \text{ um}^2$ . Folglich entspricht ein Punkt von 10 um Durchmesser einer Flächendeckung von 0,28 %, also  $79 : 27\,556 \times 100$ . Diese Flächendeckung bewirkt also praktisch keine Einschränkung des Tonwertumfangs. Bei einem 200er Raster entspricht ein Punkt von 10 um<sup>2</sup> Durchmesser schon einer Flächendeckung von 3,16 %. Hier kann man bereits von einer deutlichen Einschränkung des Tonwertumfangs sprechen, zumal dieser Effekt auch bei den tiefsten Tonwerten auftritt. Bei den tiefen Tonwerten geht es um die freien Stellen ("weiße Punkte"), die gerade noch frei bleiben. Erfahrungsgemäß sind sie noch erheblich empfindlicher als die ersten druckenden Punkte und schränken somit den Tonwertumfang von unten her noch mehr ein. Ein 200er Raster ist zugegeben ziemlich exotisch. Im Flexodruck hingegen fällt laut geplante Testelement bei einem 32er Raster bereits der Ton ab etwa 1,5 % Flächendeckung weg. Dies entspricht einem Punktdurchmesser von 22 um. Wollte man einen 60er Raster im Flexodruck verwirklichen, würde schon ein Punkt mit 5 % Flächendeckung nicht mehr gedruckt. In den Tiefen geht man im 48er Raster davon aus, daß bereits 90 % Flächendeckung als Vollfläche ausdrucken. Das heißt, bei einem 60er Raster würde bereits alles über 84% Flächendeckung als Vollton ausdrucken. Im 60er Raster würden also nur Töne von 5 % bis 84% Flächendeckung im Film ausdrucken. Durch nicht optimalen Druck ist das Ergebnis meistens noch viel schlechter. Man kann also durchaus im Flexodruck mit Rasterweiten von 60 Linien /cm und mehr drucken, aber nur um den Preis eines geringeren Tonwertumfangs.

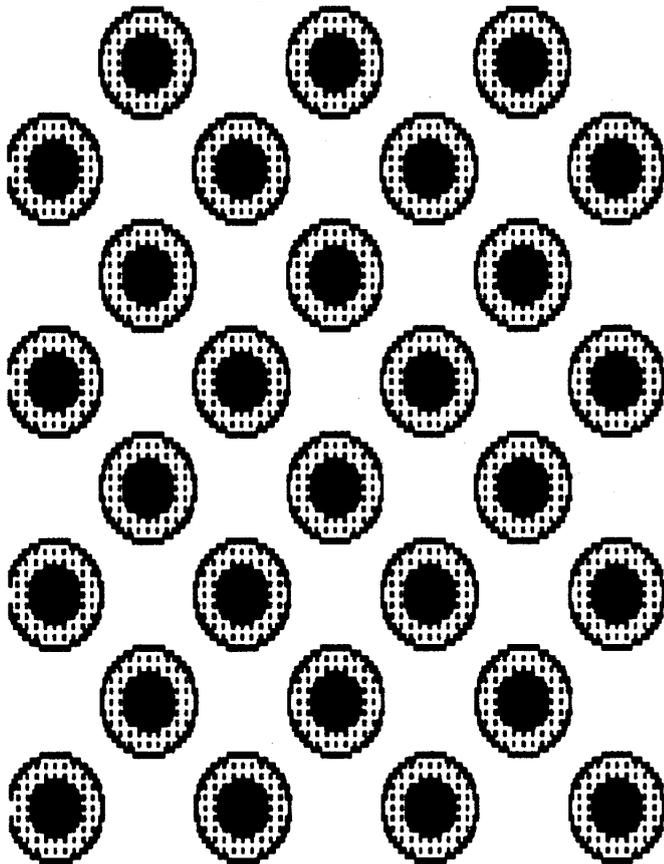
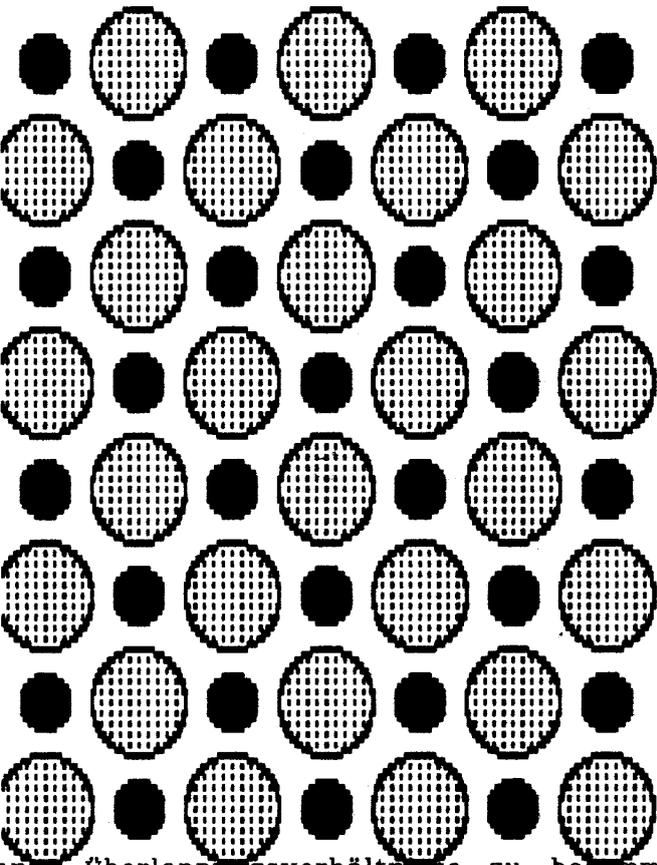
Die Lösung wäre hier, in den feinen Lichterbereichen nicht nur kleinere, sondern auch weniger Punkte zu drucken. Also eine Art frequenzmodulierter Raster in den Lichter- und in den Tiefenbereichen. Bei regelmäßigen Rastern aber einfach an bestimmten Stellen Punkte wegzulassen, führt zu schwer vorhersehbaren Moireeffekten. Dazu aber in einem späteren Kapitel mehr.

## 1.2. Ungewollte Erscheinungen im Über-einanderdruck

Die spezifischen Probleme des Übereinanderdrucks von vierfarbigen Vorlagen resultieren alle aus der einfachen Tatsache, daß Farben übereinander gedruckt anders wirken als Farben nebeneinander gedruckt. Dieser Effekt wird zwar schon erheblich dadurch gemindert, daß die üblichen Druckfarben lasierend

sind, er ist aber noch nicht beseitigt. Man könnte sich zwar entsprechende Farben vorstellen, diese sind aber noch längst nicht verwirklicht. Und es ist sehr fraglich, ob sie es jemals werden. Zu dieser einfachen Problematik siehe die dazugehörigen Skizzen. Die dargestellte Form könnte man auch als die extremste Art der Farbdrift interpretieren.

Deshalb verstärkt sich der störende Eindruck von Rasterstrukturen noch beim Übereinanderdruck von mehreren Rastern im Vierfarbendruck. Um den völligen Übereinanderdruck zu vermeiden und einigermaßen konstante



Überlappungsverhältnisse zu bekommen,

winkelt man die Raster. Beim üblichen Raster werden drei identische, rechtwinklige Raster in 30°- oder 60°-Abständen gewinkelt. Der Raster für die gelbe Farbe wird auf 15° gewinkelt. Dabei entsteht eine Struktur, die man Rosette nennt. Diese Struktur nimmt das Auge als solche wahr, das heißt, der wahrgenommene Raster erscheint niedriger auflösend als die einzelnen Raster aus denen er besteht. Nach einer Studie von Thomas Martin Schnitzler in FOGRA-Mitteilungen Nr. 141/142 ist der wahrgenommene

Raster um Wurzel aus 2 größer als die Einzelraster. Das heißt, ein 60er Raster wird als 43er Raster wahrgenommen. Erst ein 84er Raster wird als 60er Raster wahrgenommen. Nach Dr. Ing. Johannes Schoppmeyer muß man hier noch zwischen zwei Rosetten unterscheiden, einerseits der Rosette mit Mittenpunkt und andererseits der Rosette ohne Mittenpunkt. Bei einem Raster von 60 Linien/cm ergeben sich folgende Daten (Der Polygraph 8-83, Seite 496ff): Rosette ohne Mittenpunkt am inneren Ring 0,35 mm, am äußeren Ring 0,97 mm Rosette mit Mittenpunkt am inneren Ring 0,58 mm, am äußeren Ring 1,05 mm. Das bedeutet aber, daß im Extremfall, der aber in jedem Vierfarb-Offsetdruck ein paarmal auftritt, eine Struktur von der sechsfachen Rasterweite auftritt. Der äußere Ring hat zwar nur einen schwachen Kontrast, aber es bleibt immernoch der innere Ring mit einem hohen Kontrast, der etwa die dreifache Größe der Rasterweite hat. Berücksichtigt man nun, daß dieser Kontrast aber immernoch geringer ist, als der Kontrast des eigentlichen Rasters zum Papierweis, dann bewegt sich diese Feststellung doch wieder auf die ursprüngliche Feststellung von der Wuzel-zwei-fachen Rasterweite zu.

Dadurch, daß übereinander gedruckte Farben anders aussehen als nebeneinander gedruckte, kommt es unter Umständen zur sogenannten Farbdrift. Bei der üblichen Winkelung tritt sie besonders ein, wenn die Winkel nicht exakt eingehalten werden. Nach Dr. Ing. Johannes Schoppmeyer führt bereits eine Abweichung der Winkelung um 0.8' ( Bogenminuten ) bei einem 60er Raster zu wahrnehmbaren Störungen. Diese Anforderung an die Winkelung ist allein wegen Dehnungen im Papier nicht einzuhalten.

Größenordnungsmäßig in der Mitte zwischen beiden Effekten liegt das Moire. Man erhält ein typisches Moire, wenn man ein herkömmliches Raster unter zum Beispiel 5° verwinkelt. Im Druck macht sich ein Moire als in Mustern auftretende Farbschwankung bemerkbar. Es entsteht in den Vorstufen oder durch massive Störungen im Papiertransport. Besonders die Genauigkeit von Laserbelichtern reicht manchmal nicht aus, moirefrei zu belichten. Das rührt von den Grenzen des Belichters her, von unzureichender Justage oder von Filmverzug.

## 1.3. Bisherige Lösungsvorschläge

### 1.3.1. OPTI-Rastersystem

Dr. Ing. Johannes Schoppmeyer versuchte mit Hilfe von Computern unterschiedliche Winklungen und Rasterweiten zu kombinieren, um so einerseits die Farb-

drift und die Rosetten zu verringern und andererseits die Toleranz für die Winkelgenauigkeit der Montage zu erhöhen (Aufstellung der Rasterweiten und Winkel nach Schoppmeyer aus Rasterungstechnik Seite 71):

| Farbe   | Rasterweite     | Winklung |
|---------|-----------------|----------|
| Schwarz | 64,52 Linien/cm | 75°      |
| Magenta | 55,56 Linien/cm | 51,5°    |
| Cyan    | 54,05 Linien/cm | 8°       |
| Gelb    | 57,14 Linien/cm | 27°      |

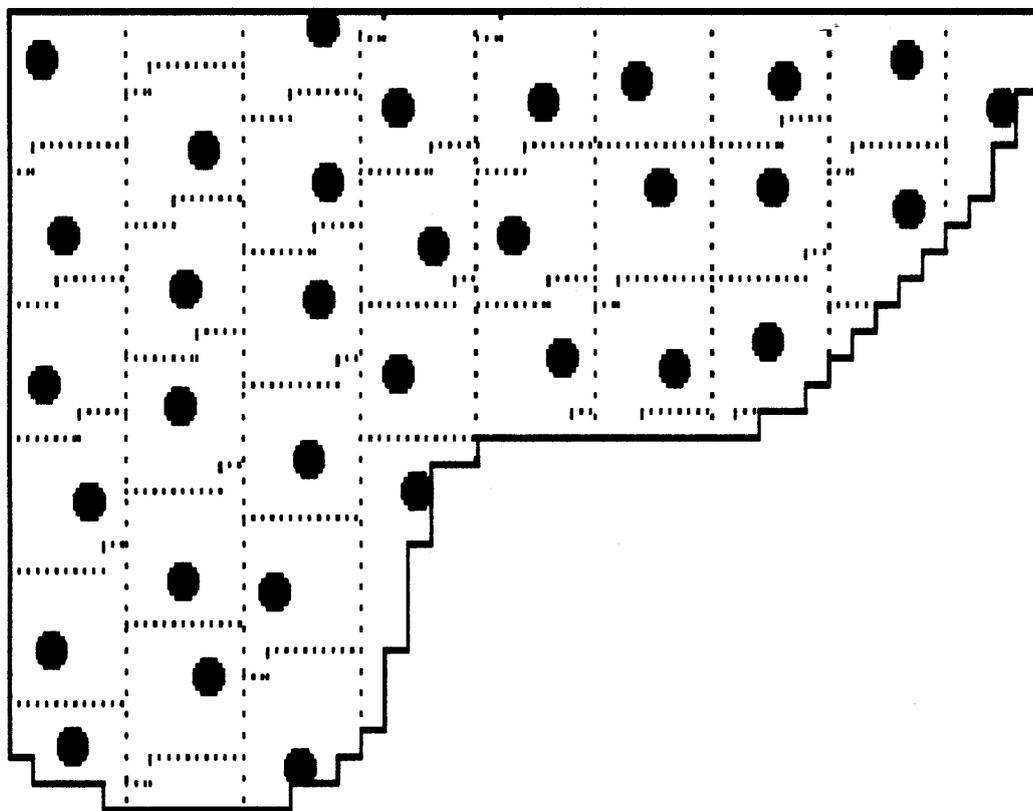
Das Problem der Farbdrift wurde gelöst, da der Raster kein ins Unendliche verwinkeltes Moire enthält. Außerdem liegt die Montagetoleranz bei  $\pm 2^\circ$ . Nicht gelöst ist das Problem der Rosette. Das wirkt sich beim Offsetdruck nicht so extrem aus, da der kleinste noch druckende Punkt relativ klein ist. Somit ist auch die Rosette klein. Bei anderen Druckverfahren wie zum Beispiel Siebdruck oder Flexodruck ist dieser kleinste noch druckende Punkt wesentlich größer. Wenn also die Rosettenstruktur verringert oder beseitigt werden könnte, wäre dies hauptsächlich in letzteren Druckverfahren eine Qualitätssteigerung. Um das OPTI-Raster ist es aber sehr still geworden. Trotz einiger Bemühungen war keine Literatur zu finden und somit auch keine Beispiele.

### 1.3.2. Frequenzmodulierter Raster

Der Bildaufbau ist beim frequenzmodulierten Raster generell anders als beim herkömmlichen Raster. Beim normalen Raster werden an festen Orten Punkte gesetzt, die je nach Grauwert, den sie repräsentieren, größer oder kleiner sind. Beim frequenzmodulierten Raster werden nur Punkte mit konstanter Größe gedruckt, jedoch werden je nach gewünschtem Grauwert mehr oder weniger Punkte gedruckt.

Bei den ersten frequenzmodulierten Rastern wurde einfach aus dem Datenbestand ein Grauwert herausgegriffen. Dieser wurde einem Quadrat in der Ausgabe zugeordnet. Über die Fläche dieses Quadrates wurden nun je nach Grauwert eine bestimmte Zahl der kleinsten noch druckenden Punkte zufällig verteilt. Die Druckergebnisse waren nicht viel besser als bei herkömmlichen Rasterverfahren, da die eigentliche Auflösung nur der Auflösung einer Grauzelle entsprach. Aus diesen ersten Versuchen stammt vermutlich die Offenlegungsschrift beim Deutschen Patentamt P 29 31 098.3-51. Anmelder ist die Dr.-Ing. Rudolf Hell GmbH mit den Erfindern Prof. Dipl.-Ing. Karl Scheuter sowie Dipl.-Ing. Gerhard Fischer, von dem auch die einzig nennenswerte Veröffentlichung zu diesem Thema stammt, nämlich das Buch "Der Frequenzmodulier-

te Bildaufbau - ein Beitrag zum Optimieren der Druckqualität", erschienen im Polygraph Verlag. In diesem Buch findet man auch einige Druckbeispiele. Eine Weiterentwicklung des frequenzmodulierten Rasters findet man in der Europäischen Patentschrift 0 074 422. Hier wird nicht einfach eine Quadratische Grauzelle angenommen, sondern das ganze Bild wird analysiert und in kleine Flächen mit ähnlichem Grauwert aufgeteilt. Die Flächen können höchst unterschiedliche Formen haben. Innerhalb dieser Flächen werden die Rasterpunkte nicht einfach zufällig angeordnet, sondern sogar möglichst regelmäßig, aber dennoch zufallsgesteuert angeordnet. Beiliegende Skizze entspricht weitgehend einer Skizze aus der Patentschrift. Sie zeigt die Umrisse einer angenommenen



unregelmäßigen Fläche mit etwa gleichem Grauwert. Die Stufen an den Rändern repräsentieren die idealisierten kleinsten Abtastquadrate. Die annähernd quadratischen Flächen innerhalb der gesamten Fläche werden vom Computer berechnet, mit der Bedingung, sie sollen möglichst gleich groß sein. Innerhalb jeder dieser Flächen wird ein Punkt gesetzt, mit der Bedingung, daß der Punkt nicht am Rand anstoßen soll. Innerhalb eines gewissen Bereiches wird die Lage des Punktes mit einem Zufallsgenerator variiert. Die Punkte folgen in weitem Maß den Konturen des Motivs. Das Bild muß vor der Rasterung genau analysiert werden. Das braucht enorme Rechnerkapazitäten. Auch die Berechnung

der Flächen, innerhalb derer der Punkt aufgezeichnet wird, ist relativ aufwendig. Wenn sich die Rechenleistung von Computern wie bisher aber weiterverbilligt, ist es für den Offsetdruck wohl das Verfahren der Zukunft, wo höchste Qualität verlangt wird. Im DTP-Bereich ist es aber wegen vorhandener Rechnerprobleme noch nicht anwendbar.

Nichts desto trotz, die Qualität der Bilder ist bestechend. Als Beispiel sei hier noch einmal auf die Arbeit von Dipl. Ing. Gerhard Fischer verwiesen, in der sich auch einige Beispiele befinden. Die Probleme bei der Wiedergabe solcher Raster liegen nicht, wie man zunächst vermutet, im Druck, sondern in der Plattenkopie. Hier muß nahezu staubfrei gearbeitet werden. Außerdem müssen Belichtungsstufen und Entwicklungszeiten peinlich genau eingehalten werden (Siehe dazu FOGRA-Mitteilungen Nr. 141/142). Aber es funktioniert und die Ergebnisse dürften den Aufwand in einigen Aufgabenbereichen rechtfertigen. Die Fachwelt war nach den ersten Versuchen sehr skeptisch. Inzwischen aber ist diese Rasterart der verbesserten Version in der Fachwelt anerkannt. Bei Druckverfahren wie eben Flexo- oder Siebdruck ist aber der kleinste druckende Punkt zu groß, so daß die Ergebnisse schlechter sind als mit herkömmlichem Raster. Zur Begriffsklärung sei hier noch angefügt, daß hier die Häufigkeit des Auftretens der Punkte, also die Frequenz, den Grauton im Sehzentrum erzeugt. Bei einem herkömmlichen Raster bleibt die Häufigkeit des Auftretens der Punkte konstant, aber die Ausdehnung dieser Punkte variiert und bestimmt dadurch den Grauwert im Sehzentrum. Man kann also sagen, daß über eine konstante Punktfrequenz eine Amplitude, in diesem Fall Dunkelstufe, moduliert wird. Deshalb kann man im Gegensatz zum frequenzmodulierten Raster auch vom amplitudenmodulierten Raster sprechen.

### 1.3.3. Sonstige

Schon als noch photomechanisch gerastert wurde, gab es den Versuch, kleinere "Zwischenpunkte" mit einzuführen, um glattere Tonwertübergänge zu erhalten. Man sprach von einem sogenannten Tripletspunkt. Auch im digitalen Bereich wäre dies möglich. Da aber die Druckergebnisse früher nicht so viel besser waren als ohne Tripletspunkt, ist diese Richtung noch nicht weiter verfolgt worden.

Ein weiteres Raster, das sich anböte, wäre ein Kornraster. Wenn man aber ein frequenzmoduliertes Raster kennt, dann kann man hier allenfalls von einem mäßigen Abklatsch sprechen. Hier findet nämlich keine Bildanalyse statt, sondern das Bild wird einfach spaltenweise abgetastet und geschrieben.

Dabei bleibt dann auch unberücksichtigt, ob sich Punktanhäufungen ergeben. Folglich gibt es dabei Punkte in verschiedensten Größen. Da sich aber der Punktzuwachs aus der Kantenlänge eines Punktes im Verhältnis zur Punktfläche ergibt, hat man hier natürlich viele Punktzunahmen überlagert, so daß man nur schwer eine klare Aussage machen kann. Die Ergebnisse scheinen schlecht zu sein, denn möglich ist dieses Raster auf vielen Scannern, es wurde aber bisher nur als Effektraster benutzt.

Schließlich ist noch ein reines Linienraster denkbar. Auf einem Plakat mit einer sehr groben Auflösung macht es sich sehr gut. Physikalisch wäre auch das Problem der Winkelung leichter zu lösen, aber das Problem ist die Verdruckbarkeit, da hier die dünnsten möglichen Linien einen zu hohen Tonwert haben. Der Tonwertumfang wäre also sehr eingeschränkt, außerdem wäre so ein Raster sehr anfällig gegen Schieben und Dublieren.

## 2. Genaue Problemanalyse

### 2.1. Bisheriger Stand

Im Kapitel über den Zusammenhang zwischen Rasterfeinheit und Tonwertumfang wurde bereits eine Beziehung von Einflußgrößen genauer behandelt. Im wesentlichen gibt es hier aber die physikalische Vorgabe des kleinsten sicher druckenden Punktes und der kleinsten noch sicher offenen Freifläche. Anhand dessen muß man sich entscheiden, welchen Tonwertumfang man braucht und welche Rasterweite noch akzeptabel ist. Das sind aber keine rein objektiven Größen mehr, sondern auch Geschmackssachen. Die Probleme im Vierfarbrasterdruck sind in allen Druckverfahren gleich. Im Prinzip reichen zur Beschreibung drei Stichworte: Farbdrift, auch Farbspiel genannt, Moire und Rosetten, wie in 1.2. schon beschrieben. Die Farbdrift ist wohl das am schwierigsten zu erfassende Problem. Um überhaupt Farborte von gerasterten Farben vorhersagen zu können, gibt es einen umfangreichen theoretischen Ansatz von Hans E.J. Neugebauer. Er geht davon aus, daß im Übereinanderdruck nicht nur die drei Grundfarben vorkommen, sondern auch das Papierweiß sowie kombinierte Übereinanderdrucke von zwei bis drei Farben, insgesamt also 8 Farben. Diese Farben werden dann farbmtrisch miteinander verrechnet. Für die Überlagerung der Farben geht er aber von statistischen Werten aus, wie oft welche der acht Farben in welchem Ton vorkommt. Dabei bleiben natürlich genau die Effekte von Rosette, Moire und Farbdrift unberücksichtigt. Näher interessierte seien hier auf die Dissertation von Hans E.J. Neugebauer an der TH zu Dresden von 1935 verwiesen mit dem Titel "Zur Theorie des Mehrfarbenbuchdrucks". W. Kresse benützte diese Erkenntnisse und packte sie 1982 in seiner Diplomarbeit in ein Programm namens RAFA, das ziemlich genaue Vorhersagen über Farborte im Übereinanderdruck und bei Winkelabweichungen von Montagen machen kann. Das Programm geht dabei aber nicht von den statistischen Werten aus, von denen Neugebauer ausgeht, sondern kann die genaue Farbabweichung an jedem Ort einer Montage genau angeben, indem es die Lage und Ausdehnung der einzelnen Punkte genau berücksichtigt. Mit anderen Worten kann das Farbspiel genau vorhergesagt werden. Dr.-Ing. Johannes Schoppmeyer überprüfte mittels dieses Programms den theoretischen Ansatz von Neugebauer und kam im Polygraph 17-82, Seite 1510 ff zu der Feststellung, daß Neugebauer die Farbverschiebung, die sich durch eine Parallelverschiebung des

Rasters ergibt, nicht berücksichtigt hat. Darüber hinaus variierte Schoppmeyer Rasterweiten und Rasterwinkelungen, um diese Effekte zu minimieren. Das Ergebnis wurde im Polygraph 8-83 unter dem Namen OPTI-Rastersystem veröffentlicht, wie unter 1.3.1. beschrieben. Die Vorteile scheinen aber die Hersteller nicht überzeugt zu haben, denn in der Fachpresse und in der Produktion findet sich seit langem nichts mehr darüber.

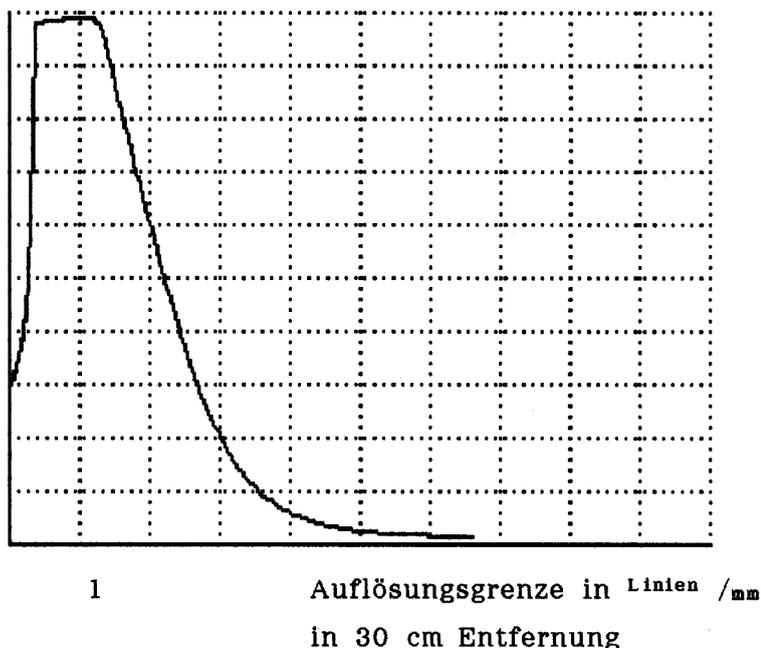
Zu Moires und Rosetten gibt es einige mathematische Formeln. Auf die Formel von D. Tollenaar soll noch näher eingegangen werden. Wobei bisher nur ungenügend untersucht worden ist, warum eine 15°-Winkelung bei Gelb nicht störend ist, warum sie aber bei dunkleren Farben zu sehr groben Moireerscheinungen führt. Dazu aber unter Punkt 2.2. mehr.

## 2.2. Zusammenhang zwischen Kontrast der Druckfarbe mit dem Hintergrund und Auflösung

### 2.2.1. Grundsätzliches

Zunächst scheint es ziemlich belanglos, daß Farben, die weniger Kontrast zum Hintergrund haben, weniger gut vom Gesichtssinn aufgelöst werden. Dieser

Kontrastempfindung 1



Zusammenhang zwischen Kontrastempfindung und Auflösungsgrenze ist noch nicht ausreichend gewürdigt worden. Aus empirischer Erfahrung hat man aber bisher bei Gelb schon ein größeres Moire in Kauf genommen. Im Druck fällt der

Effekt kaum auf, aber wenn man einmal einen Gelbfilm passergenau über einen anderen Film desselben Bildes legt, wird man ein stärkeres Moire erkennen als bei Kombinationen anderer Farben. Warum das Moire der Filme größer ist bei einer 15°-Winkelung, wie sie bei Gelb üblich ist, dazu siehe die Punkte unter 2.3. . Es wäre sehr interessant zu wissen, wie weit man diesen Effekt gezielt ausnutzen kann. Zu diesem Thema findet sich in der Fachliteratur lediglich eine Kurve nach dem Buch von Friedrich Bestenreiner "Vom Punkt zum Bild", Seite 55. Die Kurve hat aber eigentlich eine ganz andere Absicht. Sie stellt gerade den umgekehrten Zusammenhang heraus, wie man ihn an dieser Stelle eigentlich benötigt. Vielleicht wäre eine richtige Kurve aus einem medizinischen Artikel zu entnehmen, aber eine Recherche würde hier zu weit führen.

## 2.2.2. Experiment

Da also nichts Befriedigendes zu finden war, mußte eine eigene Untersuchung gemacht werden. Es mußte ein Muster, an dem man eindeutig die Auflösungsgrenze sieht, in verschiedenen Farben gedruckt werden. Ursprünglich dachte ich an Siemenssterne, fand aber dann beiliegende Testelemente wesentlich sinnvoller. Von der Testperson kann die Ziffer unmittelbar abgelesen werden der das Feld beigeordnet ist, das er gerade nicht mehr auflösen kann. Die Ziffer bezeichnet die Anzahl der Linien pro Millimeter. Der Betrachtungsabstand sollte möglichst gleich sein. Da aber der Kopf nur schwer zu fixieren ist und da einige Menschen Sehfehler haben, ist es besser, keine absolut vergleichbaren Testergebnisse bei den einzelnen Personen zu erhalten, sondern es genügt, das relative Verhältnis der einzelnen Werte bei verschiedenen Farben zu ermitteln, indem man die einzelnen Ergebnisse normiert. Die Testdrucke wurden unter 45° aufgehängt, da senkrechte und waagerechte Linien vom Gesichtssinn stärker wahrgenommen werden. Es sind auch Farben im Übereinanderdruck vorhanden, die Werte dafür wurden zwar ermittelt, jedoch nicht weiter ausgewertet. Die Beleuchtung war indirektes Tageslicht. Eine Normbeleuchtung wäre zu aufwendig gewesen. Die Tests sind in ihrer zeitlichen Abfolge aufgelistet. Ursprünglich waren noch Alter und Geschlecht der Versuchspersonen angegeben, aber hier ergaben sich keine signifikanten Korrelationen. Hier die Testergebnisse sowie die berechnete Gelbabweichung.

| Testnummer | Schwarz | Gelb | Magenta | Cyan | Gelbabweichung |
|------------|---------|------|---------|------|----------------|
| 1          | 1.8     | 1.4  | 2.0     | 2.0  | 1,38           |
| 2          | 1.8     | 1.4  | 2.0     | 2.0  | 1,38           |

---

|    |     |      |     |     |      |
|----|-----|------|-----|-----|------|
| 3  | 1.6 | 1.4  | 1.8 | 1.6 | 1,19 |
| 4  | 2.0 | 1.4  | 1.8 | 1.8 | 1,33 |
| 5  | 2.2 | 1.25 | 2.0 | 1.8 | 1,60 |
| 6  | 2.2 | 1.6  | 2.2 | 2.2 | 1,37 |
| 7  | 1.8 | 1.4  | 1.8 | 1.8 | 1,29 |
| 8  | 2.0 | 1.6  | 2.0 | 2.0 | 1,25 |
| 9  | 2.0 | 1.6  | 1.8 | 2.2 | 1,25 |
| 10 | 1.6 | 1.25 | 1.8 | 1.8 | 1,39 |
| 11 | 1.6 | 1.0  | 1.6 | 1.8 | 1,67 |
| 12 | 2.0 | 1.4  | 1.8 | 2.0 | 1,38 |
| 13 | 2.0 | 1.1  | 1.8 | 2.0 | 1,76 |
| 14 | 1.8 | 1.25 | 1.6 | 1.6 | 1,33 |
| 15 | 2.0 | 1.4  | 1.8 | 1.8 | 1,33 |
| 16 | 1.6 | 1.25 | 1.8 | 1.6 | 1,33 |
| 17 | 1.6 | 1.25 | 1.8 | 1.6 | 1,33 |
| 18 | 2.2 | 1.4  | 2.2 | 2.2 | 1,57 |
| 19 | 1.8 | 1.4  | 1.6 | 1.8 | 1,24 |
| 20 | 1.6 | 1.25 | 1.6 | 1.6 | 1,28 |
| 21 | 1.8 | 1.4  | 1.8 | 1.6 | 1,24 |
| 22 | 1.6 | 1.25 | 1.8 | 2.0 | 1,44 |
| 23 | 2.0 | 1.1  | 1.6 | 1.4 | 1,52 |

Wie kommt man nun zu der Gelbabweichung? Die Werte bei allen Farben außer Gelb schwanken innerhalb einer Testperson nur wenig. Bei verschiedenen Testpersonen ist die Reihenfolge zudem verschieden. Einheitliche Tendenz ließ sich nicht ablesen, deshalb bildet der Mittelwert über Schwarz, Magenta und Cyan den Grundwert für die Normierung. Der Wert von Gelb wird durch diesen Grundwert geteilt, also Gelbabweichung = Gelb : ((Schwarz + Magenta + Cyan) : 3) . Der Durchschnitt aus allen Gelbabweichungen beträgt 1,38. Da das Gelb beim Druck etwas verschwächt wurde und die Testpersonen sich oft überwinden mußten, den stark abweichenden Gelbwert anzugeben, kann man also 1,4 annehmen. Das bedeutet also, daß eine Struktur aus gelber Farbe 1,4 mal größer sein darf als dieselbe Struktur in einer anderen Druckfarbe der Europaskala, um gleich auffällig zu sein. Die Farben der Europaskala außer Gelb verhalten sich bezüglich des Auflösungsverhaltens in etwa gleich. Das heißt, die Gelbform kann 1,4 mal größer gerastert sein als die übrigen Formen, ohne größer auszusehen. Wenn man dies nun noch mit den Ergebnissen unter den Punkten 2.3. kombiniert, dann steht der tatsächlichen Anwendung nichts

mehr im Wege.

Die Dichten der einzelnen Farben wurden, bezogen auf das Papierweiß, durch den V-lambda-Filter ausgemessen, der in etwa die vom Menschen wahrgenommene Helligkeitsempfindung nachahmt. Schwarz wurde mit Dichte 2.81 gemessen, Cyan mit 0.83, Magenta mit 0.63 und Gelb mit 0.09. Das Papierweiß lag bei Dichte 0.07. Damit kann man über einige Berechnungen einen Vergleich mit der oben angegebenen Kurve ziehen. Der Kontrast mit dem Papierweiß wird einfach aus der Dichte zurückgerechnet und zwar mit:  $1 - 10^{-Dichte}$ . Das " $^{\wedge}$ " steht für "hoch". Das ist zumindest eine ganz gute Annäherung an den Kontrast. Damit ergeben sich folgende Kontraste: Schwarz, 1,0; Cyan 0,85; Magenta 0,77; und schließlich Gelb, 0,19. Trägt man ein Größenverhältnis von 1,4 in die Zeichnung ein, entweder, indem man von Gelb, also 0,19 aus durch 1,4 teilt, oder von einem Mittelwert der drei dunklen Farben aus multipliziert, so liegt man größenordnungsmäßig ganz gut bei der Kurve, wobei man aber Abszisse und Ordinate vertauscht hat. Eine weitere Zeichnung soll versuchen, dies zu verdeutlichen. Auch nach Aussagen des Autors ist diese Kurve aber nur als sehr vage anzunehmen, von daher kann man von einem guten Ergebnis des Tests sprechen.

## 2.2. Analyse von Rosette, Moire und Farbdrift

### 2.2.1. Einführung

Rosette, Moire und Farbdrift werden meistens als getrennte Erscheinung behandelt. Dies ist nicht sinnvoll, denn es handelt sich nur um verschiedene Auswirkungen desselben Problems. Diese Aussage wird bei Vertretern der bisherigen Betrachtungsweise auf Widerspruch stoßen. Hier soll versucht werden, obige Aussage durch "Plausibilitätsbeweise" zu untermauern. Zunächst ein Gedankenexperiment: Man stelle sich ein Runzelkornraster vor, bei dem im Druck die Gradationskurven korrekt wiedergegeben werden. Man verwendet nun dieses Raster für einen Vierfarbdruck. Eine regelmäßige Rosettenstruktur wird man vergeblich suchen. Vielleicht ergeben sich an einigen Stellen Rosetten, aber sie fallen als Einzelercheinungen kaum auf. Wenn man die Filme vor dem Druck übereinanderlegt genau einpaßt und schließlich einen oder mehrere Filme leicht verdreht, dann nimmt zwar die Schärfe des Motivs ab, aber es entstehen keine überlagerten Strukturen. Wenn man dasselbe mit einem herkömm-

lichen Raster macht, ergeben sich bekanntermaßen deutliche Moireerscheinungen. Auch bei leichtesten Verschiebungen der Filme zueinander kann es beim Runzelkornraster nicht dazu kommen, daß ein signifikanter Teil der Punkte direkt übereinander liegt. Es wird statt dessen einen bei jeder Winkelung zueinander weitgehend konstanten Anteil von Punkten geben, die übereinander liegen. Folglich tritt auch keine Farbdrift auf. Rosette, Moire und Farbdrift treten also bei diesem Beispiel nicht auf. Dies legt den Verdacht nahe, daß die drei Effekte nur auftreten können, wenn das Raster regelmäßig strukturiert ist.

## 2.2.2. Überlegungen zur Formel nach Tollenaar

### 2.2.2.1. Formel nach Tollenaar

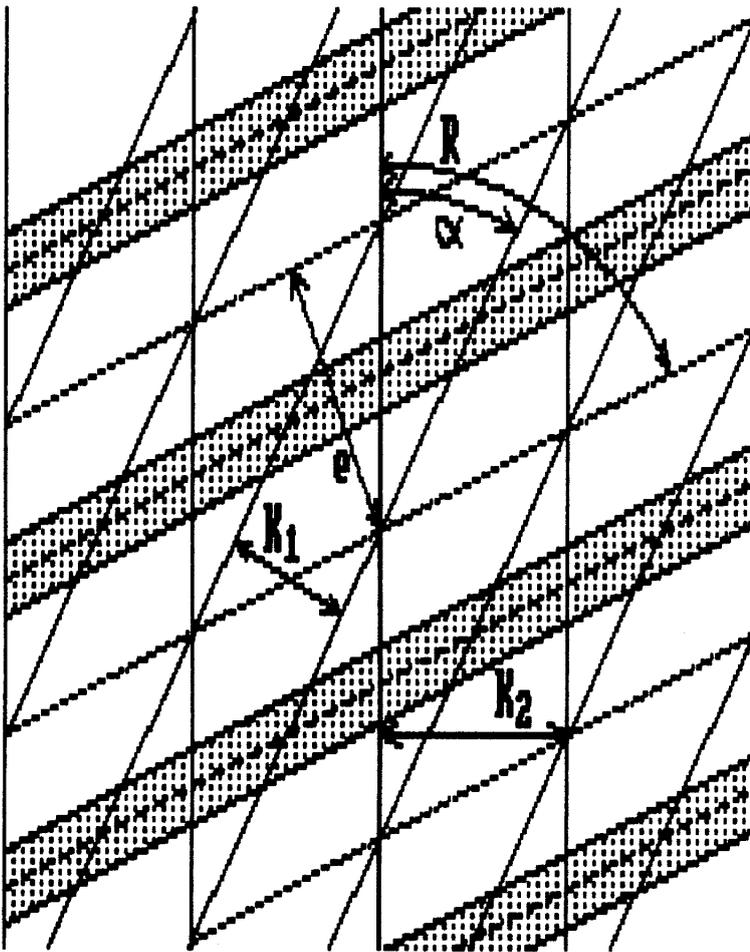
Wir betrachten nun die Formel zur Berechnung von Moires nach D. Tollenaar. Sie betrachtet nur Linien. Die Formel geht also von zwei Linienstrukturen mit jeweils regelmäßigem Abstand aus. Die beiden Strukturen werden nun in einem Winkel überlagert. Es entsteht eine neue Linienstruktur, die aber nur bei genauerer Betrachtung sichtbar wird. Ihr Kontrast zum Hintergrund ist geringer als derjenige der ursprünglichen Linienstrukturen. Der Winkel und die Weite der neuen Linienstruktur ergibt sich nach der Formel eindeutig aus den jeweiligen Winkeln und Frequenzen der ursprünglichen Linienstrukturen.

Die Formel für den Abstand ( e ) der neuen Linienstruktur lautet:

$$e = \frac{k_1 \cdot k_2}{\sqrt{k_1^2 + k_2^2 - 2 \cdot k_1 \cdot k_2 \cdot \cos \alpha}}$$

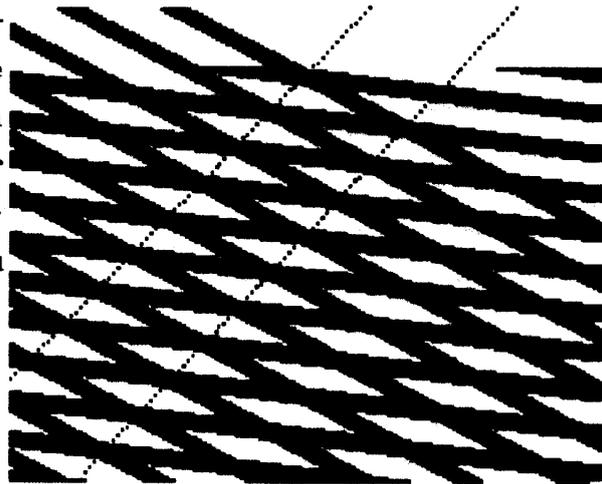
Die Formel für den Winkel ( R ) der neuen Linienstruktur lautet:

$$R = \alpha + \arctan \frac{k_1 \cdot \sin \alpha}{k_2 - k_1 \cdot \cos \alpha}$$



Die Bezeichnungen in den Formeln beziehen sich auf die Bezeichnungen in der zugehörigen Skizze. Die Formeln und die Skizze sind dem Buch "Rasterungstechnik fotomechanisch und elektronisch" von Dieter Morgenstern, Seite 57 entnommen. Dort findet man auf den folgenden Seiten auch einige interessante abgeleitete Formeln für herkömmliche Raster. Die Formel ist nur für Linien geometrisch eindeutig beweisbar. Die Skizze mit den dicken Linien soll demonstrieren, wie man die Formel

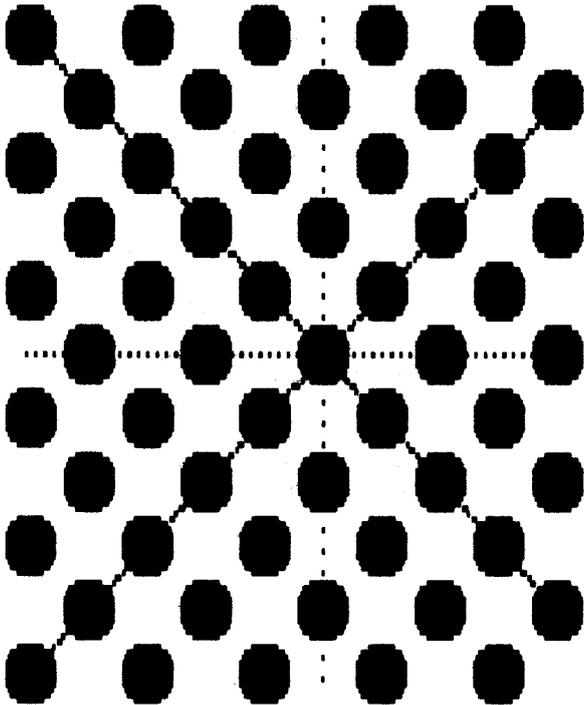
anhand der Praxis überprüfen kann. Die gestrichelten Linien repräsentieren dabei die neu entstandene Linienstruktur, allerdings um eine halbe Phase versetzt, aus graphischen Gründen. Was hat diese Formel aber mit den Rosetten, Moires und Farbdriften in einem Punktraster zu tun?



### 2.2.2.2. Folgerungen aus der Formel

Das Auge ergänzt in Linie angeordnete Punkte zu einer Linie. Diese Erfahrung

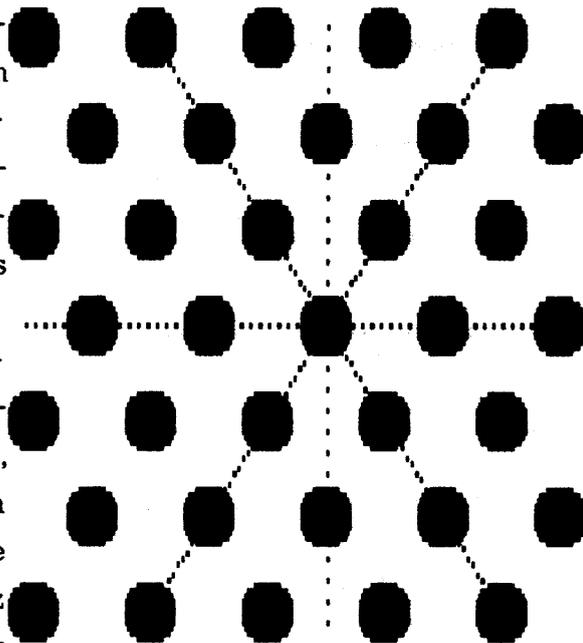
macht man immer, wenn man ein regelmäßiges Raster aus einer Entfernung ansieht, die es dem Auge erlaubt, die einzelnen Punkte noch aufzulösen. Die Betrachtung eines solchen Rasters aus entsprechendem Abstand ist für den Menschen sehr unangenehm, da sich das Sehzentrum nicht entscheiden kann, ob nun zwei Linienstrukturen oder ein Grauwert vorliegen. Also "springt" das Sehzentrum zwischen den Deutungsmöglichkeiten. Einige Menschen bekommen beim Betrachten solcher Muster Kopfschmerzen. Interessant dabei ist, daß



regelmäßige Strukturen wesentlich auffälliger sind als unregelmäßige. Dabei ergänzt das Sehzentrum jeweils die kürzesten Entfernungen zwischen jeweils zwei Punkten zu Linien. Das kommt daher, daß diese Linien den höchsten Kontrast mit dem Hintergrund bilden, da sie gleichsam am dichtesten mit Punkten besetzt sind. Folglich werden die anderen denkbaren Linien ausgeblendet. Sie sind aber nicht weg, was man bei Überlagerung von verschiedenen regelmäßigen Punktrastern schmerzhaft feststellen wird. In den entsprechenden Skizzen sind die möglichen Linien gestrichelt gekennzeichnet. Im Normalfall werden aber nur die

Linien mit der kürzesten Punktverbindung wahrgenommen. Zum Vergleich dazu die Anordnung eines idealisierten Tiefdruckrasters. Von der Punktanordnung ist es lediglich ein gewissermaßen in eine Richtung gelängtes Raster.

Was hat aber nun diese eigentliche Linienformel mit Moires bei Punktrastern zu tun? Jedes konventionelle, rechtwinklige Punktraster kann durch zwei in 90° überlagerte regelmäßige Linienstrukturen gleicher Frequenz aufgefaßt werden, bei dem eben nur



einige Punkte fehlen. Diese Auffassung läßt die nächst größeren Linien im Winkel von  $45^\circ$  außer acht, die streng genommen als dritte Linienstruktur mit einbezogen werden müßten, da die Punktdichte und somit der Kontrast nicht wesentlich geringer ist. Bei einem rautenförmigen Raster wie im Tiefdruck (siehe obige Skizze) muß man die dritte Linienstruktur in jedem Falle mit einbeziehen. Diese Betrachtungsweise ist aber in der Literatur nicht üblich, wo man nur von herkömmlichen Rastern ausgeht. Bei Überlagerung von zwei herkömmlichen Rastern gibt es demnach also sechs Kombinationsmöglichkeiten von Linienstrukturen. Da die Linienweiten immer gleich sind, erzeugen nur zwei Linienkombinationen eine nennenswert größere Struktur, die sich in einem Winkel von weniger als  $45^\circ$  überschneiden. Insgesamt entstehen nur zwei neue Linienstrukturen. Diese schneiden sich aus Symmetriegründen in einem Winkel von  $90^\circ$ . Durch die Kombination der beiden Linienstrukturen erhält man die Ausrichtung und die Größe einer üblichen Moirezelle. Wegen der speziellen Voraussetzungen ist diese Zelle quadratisch. Winkelt man so, daß diese Zelle immer größer wird, spricht man von einer Verwinkelung ins Unendliche. Diese ist der fließende Übergang zur Farbdrift. Die Farbdrift wird aber in der Praxis meistens nicht durch den Film verursacht, sondern durch den Papierverzug in der Druckmaschine, da es sich auch von der Theorie her um minimale Winkelschwankungen handelt, in der Größenordnung von einer Bogenminute. Man kann also mit Hilfe dieser Formel nur die Größe und Ausrichtung einer Moirezelle angeben, nicht deren Aussehen, da dieses auch von der Punktform und dem jeweiligen Tonwert abhängt. Wegen der vielen Einflußfaktoren, wäre eine genauere Angabe über Kontrast oder Form nicht sinnvoll. Die Formel ist also in gewissen Grenzen anwendbar, folglich ist die Betrachtungsweise, die hinter der Theorie steckt, mindestens brauchbar.

Da sowohl die Formel als auch das Gedankenexperiment gleichermaßen auf alle drei Effekte, also Rosette, Moire und Farbdrift angewandt werden können, scheint es doch sehr wahrscheinlich, daß es sich nur um eine einzige Erscheinung handelt, die lediglich bei verschiedenen Größenordnungen verschiedene Namen hat.

### 2.2.2.3. Verbesserte Anwendung der Formel

Die Formeln sind relativ kompliziert und erscheinen auf den ersten Blick sehr abstrakt. Für die weiteren Betrachtungen ist es aber sehr interessant zu wis-

sen, was passiert, wenn man einen Eingangsparameter der Formel ändert. Beide Formeln haben drei Eingangsgrößen, nämlich die Linienweite  $K_1$ , die Linienweite  $K_2$  und den Winkel  $\alpha$ , also den spitzen Winkel, den beide miteinander einschließen. Zur Betrachtung setzt man die Linienweite  $K_1$  gleich 1 und setzt für  $K_2$  das Verhältnis zwischen  $K_1$  und  $K_2$  ein, also " $K_2 : K_1$ ". Bezeichnet man diese Verhältniszahl als  $x$  und ferner die sich ergebende Linienweite als  $y$ , so hat man eine klassische Funktion  $y=f(x)$  mit dem Scharparameter  $\alpha$ . Die Funktion lautet:

$$f(x) = \frac{x}{\sqrt{x^2 + 1 - 2x \cos \alpha}}$$

Der Graph der Kurvenschar (siehe Skizzenblätter im Anhang an dieses Kapitel) gibt schnell Auskunft darüber, wie groß ein Moire wird, wenn ich bei einer gegebenen Linienstruktur die Linienweite oder den Winkel einer zweiten Linienstruktur variere und diese überlagere. Da man die Formel immer nur auf den spitzen Winkel beziehen darf, ist es nicht sinnvoll, Winkel von mehr als  $90^\circ$  mitzubetrachten. An der  $x$ -Achse trägt man einfach das Größenverhältnis der zweiten zur ersten Linienstruktur ab. Dann sucht man sich den Graph, dessen  $\alpha$  dem eigenen am nächsten kommt, oder interpoliert einen Graphen nach Augenmaß. Am zugeordneten  $y$ -Wert kann man nun erkennen, um wieviel das so entstehende Moire größer ist als die Linienweite des ersten Moires. An der Kurvenschar erkennt man bereits grob einige Dinge:

- Bei einem Winkel von  $90^\circ$  kann kein Moire im Verhältnis zur ersten Linienweite größer als 1 sein
- Bei einem Winkel bis hinunter zu  $45^\circ$  kann kein Moire größer als das 1,5-fache der ursprünglichen Linienweite sein
- Bei Winkeln unter  $45^\circ$  ist besonders ein Größenverhältnis von etwa "1 : 1" kritisch
- Ab einem Größenverhältnis von "1 : 2" bzw. "1 : 0,5" kann kein Moire größer als das 2-fache der ersten Linienweite sein
- Farbdrift, also Moire in hoher Größenordnung tritt nur bei Winkeln unter  $15^\circ$  und einem Größenverhältnis der Linienweiten gegen "1 : 1" auf

Mit Hilfe einer Extremwertberechnung kommt man zu dem Ergebnis, daß bei einem gegebenen Winkel  $\alpha$  das Moire nie größer als  $1/\cos(\alpha)$  ist, mit anderen Worten, dort liegt das Maximum der Kurve. Dieses ist dann natürlich bei  $0^\circ$  nicht definiert.

Bei der Formel für die Berechnung des Winkels der neu entstehenden Linienstruktur ist die Sache noch etwas komplizierter. Auch bei der Darstel-

lung als Kurvenschar analog zur Berechnung der Moireweite ergibt sich kein graphisch irgendwie verwertbares Bild. Abgesehen davon geht die Formel in ihrer bisherigen Form großzügig über die Vorzeichen und damit die Drehrichtung des Winkels hinweg. Deshalb ist es sinnvoll eine andere Bezugsgerade zu wählen. Bisher wird dafür einfach die Richtung einer der Linienstrukturen verwendet. Besser ist es, die Winkelhalbierende des stumpfen Schnittwinkels zwischen beiden Linienstrukturen zu wählen, oder anders ausgedrückt, die Normale zur Winkelhalbierenden von  $\alpha$ . Wenn nämlich die beiden Linienweiten gleich groß sind, dann hat die entstehende Linienstruktur immer diese Ausrichtung. Ähnlich wie oben wurden  $K_1$  und  $K_2$  durch  $x$  ersetzt. Die so entstehende Formel lautet folgendermaßen:

$$f(x) = \arctan \left( \frac{1}{\tan \frac{\alpha}{2}} \cdot \frac{x-1}{x+1} \right)$$

Die  $y$ -Achse repräsentiert die Abweichung von der Bezugsgeraden, wie sie oben beschrieben ist. Der einfacheren Lesbarkeit halber wird das Ergebnis in Grad und nicht in Bogenmaß angegeben. Die Skizze befindet sich ebenfalls im Anhang zu diesem Kapitel. Unmittelbare Aussagen ergeben sich aus der Kurvenschar keine. Man kann den Graphen jedoch wieder ähnlich benutzen wie den vorigen. Nur erhält man als Ergebnis eben die Abweichung von der Bezugsgeraden in Grad. Wenn man die Formel benützt, muß man sich vorher genau überlegen, welche Linienweite man durch welche teilt, um  $x$  zu erhalten, und welche Drehrichtung der Abweichung als positiv definiert werden soll, sonst erhält man die falsche Drehrichtung.

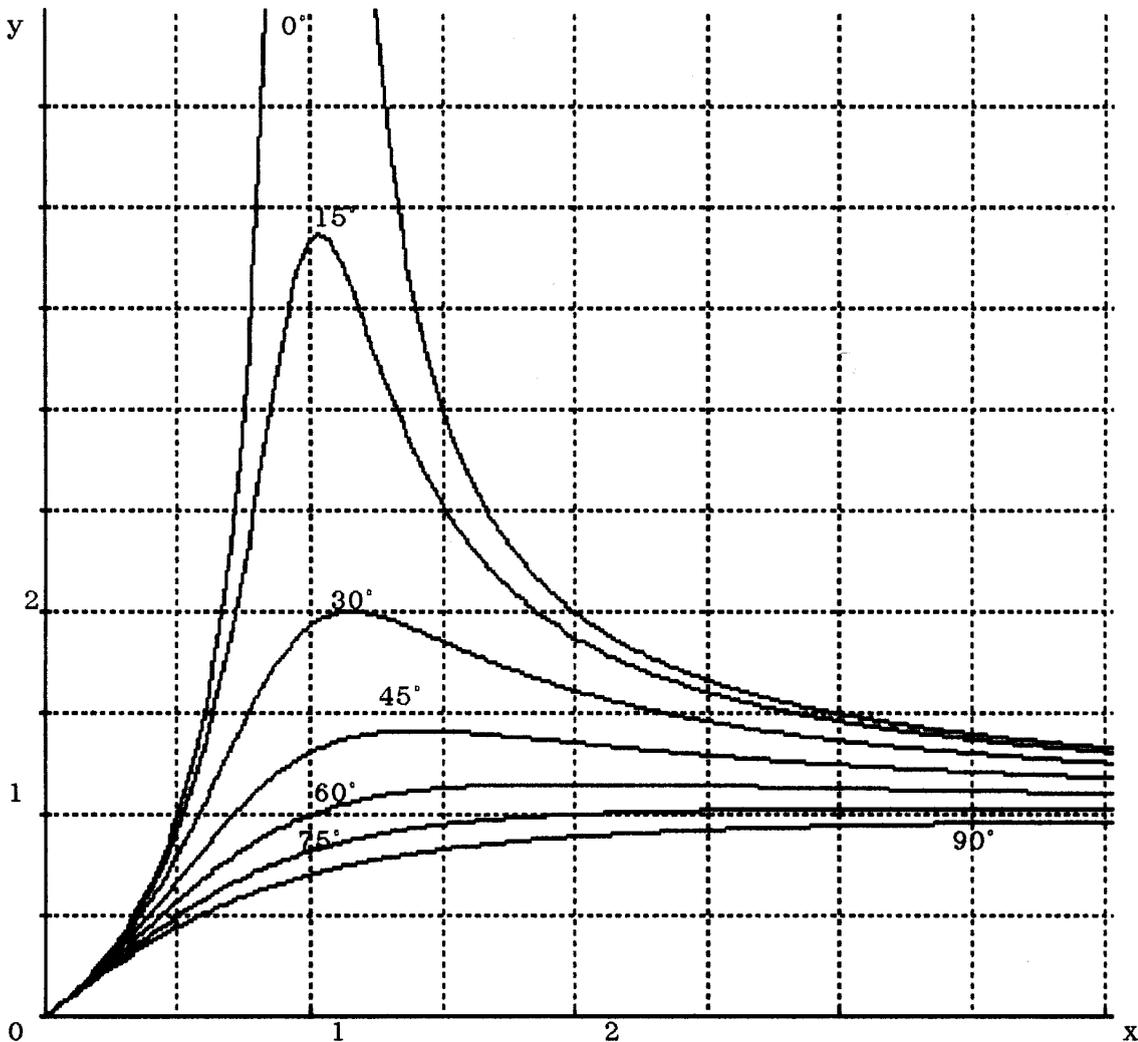
Insgesamt sollen beide Kurvenscharen vorstellbar machen, was bei der Überlagerung zweier Linienstrukturen geschieht, welche Konstellation von Parametern günstig und welche ungünstig ist. Dabei sollte man sich des Zusammenhangs zwischen Linien und Punktrastern bewußt bleiben, wie er vorhin dargestellt wurde.

#### 2.2.2.4. Beschreibung von Rastern im Hinblick auf die Formel

Ein herkömmliches Raster läßt sich durch zwei Linienstrukturen, die sich unter  $90^\circ$  kreuzen am besten beschreiben, wenn man die weniger kontrastreichen

Linien, die theoretisch noch möglich wären, einmal außer acht läßt. Bei einem Vierfarbdruck müßte man vier solcher Doppellinienstrukturen überlagern. Hier soll aber Gelb, wegen seines geringeren Kontrastes nicht mit einbezogen werden. Zur genauen Begründung dafür siehe Punkt 2.2.2. Wenn man nun jedes Raster ganz bildlich als Kreuz darstellt, wie in der Skizze im nächsten Punkt 3.1. dargestellt, dann ist klar, warum die Raster normalerweise unter  $30^\circ$  bzw.  $60^\circ$  verwinkelt werden. Auf diese Art sind sie einfach gleichmäßig verteilt. Sehen wir uns im Vergleich dazu ein graviertes (!) Tiefdruckraster an. Dieses läßt sich unter ähnlichen Voraussetzungen wie beim herkömmlichen ebenfalls durch zwei Linien darstellen, allerdings sind diese nicht unter  $90^\circ$  verwinkelt. Der Übergang von der reellen Winkelung zur bloß abstrakten Darstellung ist in einer weiteren Skizze dargestellt. Analog verläuft dieser Übergang beim herkömmlichen Raster. Wenn man diese Art der Winkelung auf ein normales autotypisches Druckverfahren wie den Offsetdruck überträgt, dann hat man insgesamt einen Freiheitsgrad in der Winkelung hinzugewonnen, allerdings nimmt auch die Übersichtlichkeit beträchtlich ab. Die ungefähre Tiefdruckrasterung ist in der Skizze im nächsten Punkt analog zum herkömmlichen Raster gezeigt. Interessant dabei ist, daß sich, solange die Überflutung noch nicht eingesetzt hat, ein Tiefdruckraster von der Geometrie her sich ähnlich wie ein normales Raster verhält und, daß diese Art von Rasterung auch für autotypische Druckverfahren denkbar wäre.

## Kurvenschar Moireweite nach Punkt 2.2.2.3.



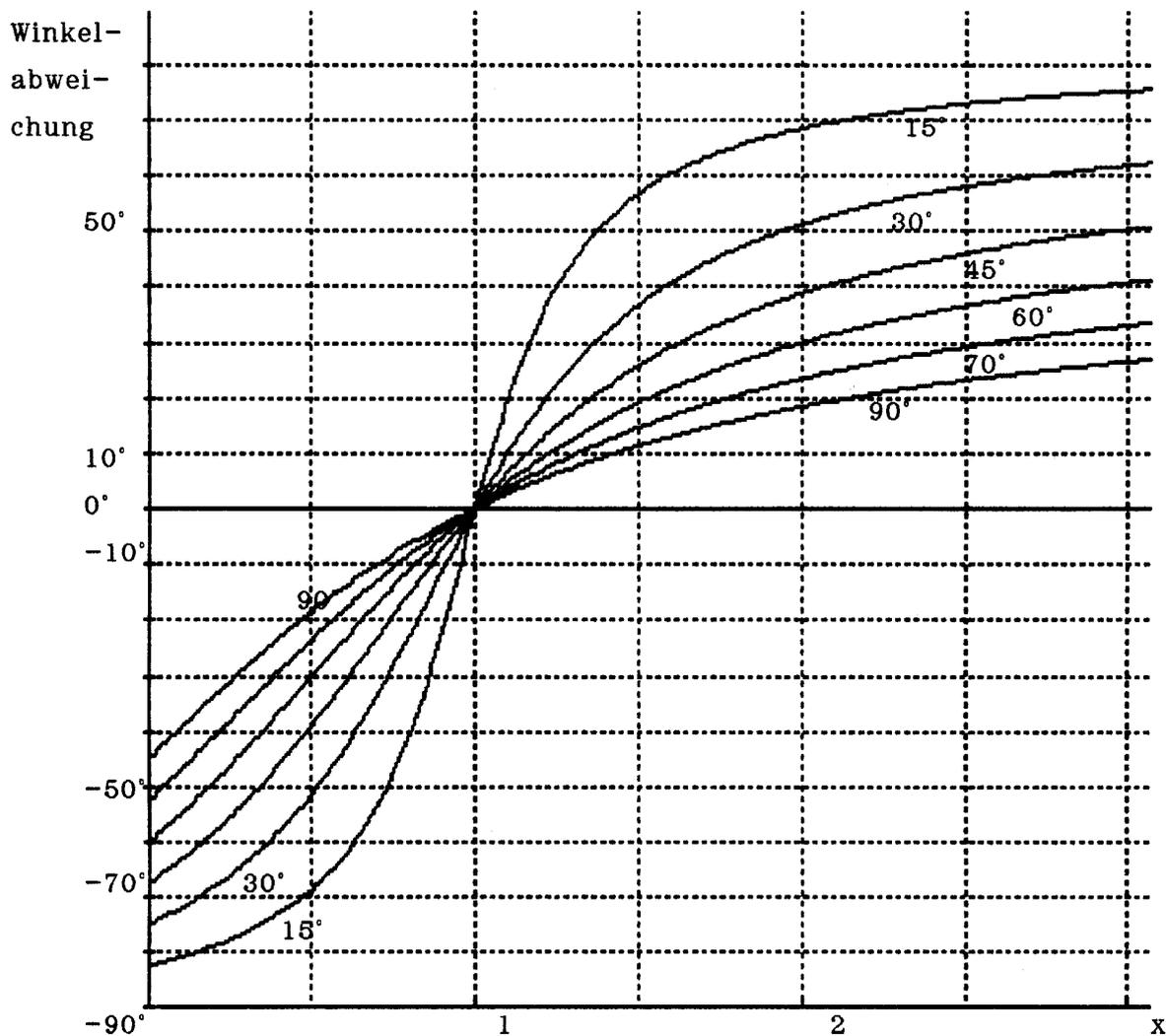
"x" bezeichnet die Relation der beiden Linienweiten zueinander, also "Linienweite 2 : Linienweite 1".

"y" bezeichnet die Relation zwischen der Weite des Moires, das bei der Überlagerung der beiden Linienstrukturen entsteht und der ersten Linienweite, also "Moireweite : Linienweite 1".

Da die Größenangaben der Linienweite und der Moireweite gleich sind, ergeben sich Zahlenwerte ohne Benennung. Die Benennung ist also letztlich egal, sie kann zum Beispiel "millimeter" oder "Linien pro centimeter" sein.

Die Kurvenschar repräsentiert jeweils den spitzen Winkel zwischen den beiden Linienstrukturen, in diesem Falle, 90°, 75°, 60°, 45°, 30°, 15° und 0°.

## Kurvenschar Winkelabweichung des Moires nach Punkt 2.2.2.3.



"x" ist das Verhältnis der Weiten der beiden Linienstrukturen, aus denen das Moire entsteht, also "Linienweite 2 : Linienweite 1".

Werden zwei Linienstrukturen gleicher Weite überlagert, also im Verhältnis 1 : 1, dann hat das entstehende Moire immer die Ausrichtung einer Normalen ( $90^\circ$ -Winkel zu einer anderen Geraden) zur Winkelhalbierenden der Linienstrukturen. Ist das Größenverhältnis ungleich eins, dann ergibt sich eine Abweichung zu dieser Normalen, diese wird in der Zeichnung als Winkelabweichung bezeichnet und ist in diesem Fall in Winkelgrad angegeben.

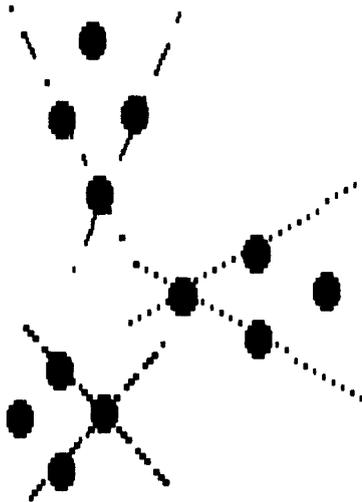
Die Kurvenschar repräsentiert den spitzen Winkel zwischen den Linienstrukturen. Der Winkel zwischen beiden Strukturen ist der zweite Eingangspa-

parameter. Er kann beliebig variiert werden. Werte über  $90^\circ$  sind jedoch nicht sinnvoll. Es können andere Werte für den Winkel angenommen werden als die angegebenen  $15^\circ$ -Schritte. Durch Interpolation kann man sie leicht annähern.

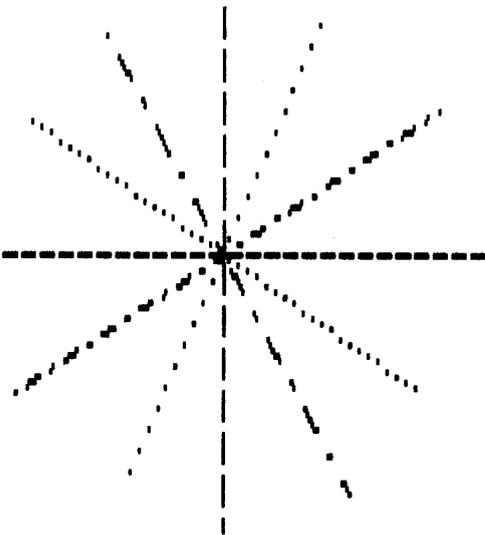
### 3. Neue Rastermöglichkeiten anhand der Geometrie der Tiefdruckgravur

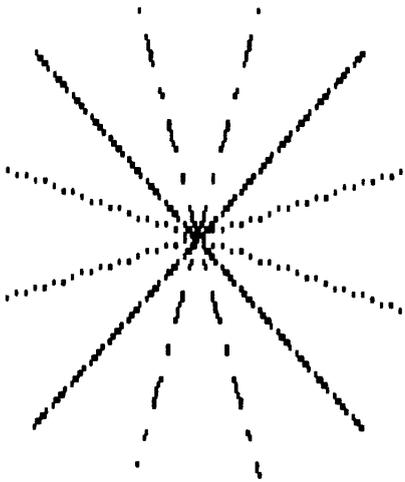
#### 3.1. Betrachtung des Tiefdruckrasters

Von der Theorie her ist es also möglich die Geometrie des Tiefdruckrasters auf ein autotypisches Raster zu übertragen. Um den Zusammenhang deutlich zu machen, sind noch einige Skizzen beigelegt. Sie zeigen die Darstellungsarten für Raster, wie sie bereits unter Punkt 2.2.2.4. besprochen wurden. Besonders interessant ist dabei die Ähnlichkeit der prinzipiellen Darstellung des Tiefdruckrasters und des herkömmlichen Rasters. Welche Vorteile soll nun eine Übertragung der Tiefdruckgeometrie auf autotypische Druckverfahren bringen? Wenn man einen Blick auf ein Tiefdruckzeugnis wirft, wird man mit bloßem Auge keine



Rosette, kein Moire und allenfalls eine Farbdrift feststellen, wenn man mehrere Exemplare zur Hand hat. Farbdrift dürfte im Tiefdruck eigentlich nicht vorkommen. In der letzten Skizze sieht man, daß eine der Rasterungen wie beim herkömmlichen Raster in sich rechtwinklig ist. Die anderen beiden sind entweder vertikal oder horizontal um 1,5 gelängt. Im letzteren Fall spricht man von gestaucht. Daß damit auch die Form variiert wird, tut hier nichts zur Sache, da für die Linienstrukturen lediglich die Anordnung der Punkte von Bedeutung ist. Die gelängten oder gestauchten Näpfchen lassen sich schneller gravieren als die feinen Näpfchen, deshalb bedienen sich viele Tiefdruckereien nicht der vom Graviermaschinenhersteller angegebenen Winkelung, sondern sie benutzen dieselbe Winkelung zweimal. Hierbei kann es zu erheblicher Farbdrift kommen. Dieses Risiko nimmt man aber in Kauf. Etwas gemildert wird dieser Effekt dadurch, daß die Tiefdruckfarbe stark verläuft. Folglich sind die einzelnen Punkte weniger randscharf als zum Bei-





spiel beim Offsetdruck. Dies wirkt sich günstig hinsichtlich sich eventuell ergebender Linienstrukturen aus. Bei einem korrekt gewinkelten und gedruckten Tiefdruckprodukt wird man die gegenüber dem Offsetdruck überlegene Qualität einräumen müssen. Diese ergibt sich aus dem oben schon erwähnten Effekt des Verfließens, durch die hohe Auflösung, aber auch durch die günstigeren Eigenschaften der Winkelung.

### 3.2. Theoretische Optimierung

Um die Qualität eines Rasters hinsichtlich Moirefreiheit bestimmen zu können, muß man erst ein Maß berechnen. Dafür eignet sich am besten die größte Moireweite, die sich aus der Überlagerung der Linienstrukturen der Raster ergibt. Wir wollen sie als primäre Moireweite bezeichnen. Die primären Moires mit den Linien oder auch mit anderen primären Moires überlagert, ergeben wieder Moires, die sekundären Moires. Durch die oben angegebenen Kurvenscharen läßt sich die primäre Moireweite bei einem herkömmlichen Raster leicht mit knapp 2 ablesen. Die sekundäre Moireweite beträgt knapp 4. Bei einem Tiefdruckraster wird die Sache aber wesentlich unübersichtlicher. Deshalb habe ich ein Auswertungsprogramm geschrieben. Die Verbindung von vier Punkten, wie sie unter 3.1. gezeichnet sind, ergibt im Tiefdruck immer eine Raute mit einer waagrechten und einer senkrechten Diagonale. Zur Eingabe in das Programm bedeutet s1 die senkrechte Diagonale der ersten Raute, w1 bedeutet die waagrechte Diagonale. s2 und w2 repräsentieren das zweite Raster und s3 und w3 schließlich das dritte. Als Ergebnis werden diese Paare nocheinmal ausgedruckt, anschließend das primäre Moire und das sekundäre Moire, aber nur dann, wenn das sekundäre Moire kleiner als doppelt so groß wie das primäre Moire ist. Das Programm ist in Amiga-Basic geschrieben. Das Umschreiben in einen anderen Dialekt oder eine andere Sprache dürfte kein Problem sein. Die Eingabe der einzelnen Werte ist eigentlich für die Anwendung nicht so interessant. Interessanter ist es, mit Schleifen bestimmte Werte durchzuvariieren, was jeder, der ein bisschen programmieren kann, ohne weiteres tun kann. Dazu müssen in Basic die "FOR"-Anweisungen vor die Zeilenmarke "Hauptprogramm:"

gesetzt werden, die "NEXT"-Anweisung vor die Zeilenmarke "Schleifenende". Beim Abtippen muß man unbedingt die Kommentare weglassen, die ich der Übersicht halber nicht in "REM"-Zeilen geschrieben habe. Hier also das Programm:

### 3.3. Computerprogramm zur Berechnung von Moires bei gravurähnlicher Anordnung

```
PI=3.141592656#
CLEAR
DEFINT i,k,l,m,n
DIM e(666), r(666)
REM sonstige Variable sind Real
```

Konstanten und Variable werden gesetzt.

```
INPUT"Wie groß ist s1";s1
INPUT"Wie groß ist w1";w1
PRINT s1;w1
INPUT"Wie groß ist s2";s2
INPUT"Wie groß ist w2";w2
PRINT s2;w2
INPUT"Wie groß ist s3";s3
INPUT"Wie groß ist w3";w3
PRINT s3;w3
ep=999999999999#
```

Eingabe der vertikalen und horizontalen Diagonalen. Das fertig definierte Raster wird angezeigt.

ep bestimmt die primäre Moiregröße, unter der eine Ausgabe erfolgt.

Hier eventuelle "FOR"s einbauen.

Hauptprogramm:

```
h=h+1
em=0
r(1)=ATN(w1/s1)
r(4)=-r(1)
e(1)=ABS((SIN(r(1))*s1))
e(4)=e(1)
r(2)=ATN(w2/s2)
```

Beginn des Hauptprogramms.

Gibt an wieviele Durchläufe.

Die Winkel und Moireweiten werden berechnet. r repräsentiert den Winkel, e die Moireweite.

```

r(5)=-r(2)
e(2)=ABS((SIN(r(2))*s2))
e(5)=e(2)
r(3)=ATN(w3/s3)
r(6)=-r(3)
e(3)=ABS((SIN(r(3))*s3))
e(6)=e(3)
l=6
eo=ep
FOR n=1 TO 5
  FOR m=n+1 TO 6
    GOSUB Kern
  NEXT m
NEXT n
IF n>9999 THEN Schleifenende:
efp=em
l=21
eo=2*efp
FOR n=1 TO 20
  FOR m=n+1 TO 21
    GOSUB Kern
  NEXT m
NEXT n
IF n>9999 THEN Schleifenende:
es=em
ep=efp
PRINT s1;w1;"*";s2;w2;"*";s3;w3;ep;es

```

Anzahl der Kombinationen.  
Abbruchbedingung, wenn ep überschritten wird.

eigentliche Berechnung der Moires

analog zur vorigen Schleife

Ausgabezeile auf Bildschirm.  
Hier eventuelle "NEXT"s einbauen.

Schleifenende:

END

REM Programmende

Kern:

Routine zur Moireberechnung.

l=l+1

rr=r(n)-r(m)

Berechnung von alpha.

IF ABS(rr)>.001 THEN

Berechnung des Nenners der

```

qnene=e(n)^2+e(m)^2-2*e(n)*e(m)*COS(rr)
IF qnene>0 THEN
    e(l)=e(n)*e(m)/SQR(qnene)
ELSE
    e(l)=eo+1
END IF
ELSEIF e(n)=e(m) THEN
    e(l)=eo+1
ELSE
    e(l)=e(n)*e(m)/ABS(e(n)-e(m))
    r(l)=0
END IF
IF e(l)>eo THEN
    m=10000
    n=10000
    RETURN
END IF
IF e(l)>em THEN em=e(l)
IF COS(rr)=0 THEN
    r(l)=PI/4*SGN(rr)
ELSE
    qnenr=(e(n)/e(m)+1)*TAN(rr/2)
    zaer=e(n)/e(m)-1
    IF NOT(qnenr=0) THEN
        r(l)=ATN(zaer/qnenr)+r(n)-rr/2+PI/2
    ELSEIF zaer=0 THEN
        r(l)=r(n)
    ELSE
        r(l)=r(n)-PI/2*SGN(rr)
    END IF
END IF
RETURN
RETURN

```

Moireweite und Prüfung auf 0.  
Berechnung der Moireweite.

Ausnahmefälle.

Abbruchbedingung.

Berechnung des Moirewinkels in  
Teilen.

Berechnung und Addition des Be-  
zugswinkels.

Ausnahmefälle.

Ende des Unterprogramms.

Ein Vorläufer des Programms sollte die optimale Kombination von Rastern mit Hilfe des Zufallszahlengenerators finden, indem es eine Art Evolution nachvollzieht. Leider ist das Programm als Interpretersprache zu langsam. Mit ins-

gesamt 6 Unbekannten ist es auch schwierig, systematisch alle Kombinationsmöglichkeiten zu testen. So mußte also eine Auswahl verschiedener Eingangsvoraussetzungen getroffen werden, die noch überschaubar genug war, sie mit dem Computer systematisch zu prüfen. Die prinzipielle Geometrie der Tiefdruckanordnung scheint dabei sehr sinnvoll.  $w_2$  und  $s_3$  werden gleich 1 gesetzt, so daß eine gelängte und eine gestauchte Raute entstehen, mit der kurzen Diagonalen gleich 1. Auf diese Weise mußten nur noch vier Größen variiert werden. Dabei in Verbindung mit noch einem weiteren Test wurde herausgefunden, daß es am günstigsten ist, wenn man die Geometrie noch mehr in Richtung Tiefdruckgravur einschränkt. Man kann nämlich  $s_1$  und  $w_1$  gleichsetzen, so daß sich ein verbleibendes quadratisches Raster ergibt, das nur in der Größe variiert. Außerdem kann man  $s_2$  und  $w_3$  gleichsetzen, so daß die gelängte und eine gestauchte Raute dieselben Proportionen haben. Hier möchte ich zwei Ergebnisse herausgreifen. Die erste Zeile mit Werten stellt meine Ergebnisse für ein gewisses Optimum dar, das auch die sekundären Moires klein hält. Allerdings ist es nur auf eine Nachkommastelle genau. Die zweite Zeile ist eine möglichst weite Annäherung an den Tiefdruck.

| $s_1$ | $w_1$ | $s_2$ | $w_2$ | $s_3$ | $w_3$ | primäres Moire | sekundäres Moire |
|-------|-------|-------|-------|-------|-------|----------------|------------------|
| 1,2   | 1,2   | 1     | 2,5   | 2,5   | 1     | 2,15           | 3,03             |
| 1,1   | 1,1   | 1     | 1,5   | 1,5   | 1     | 3,86           | 5,50             |

Man kann statt des Programms auch versuchen, sich bestimmte Kriterien herauszugreifen und die Formeln entsprechend umzustellen, so daß sich neue formale Zusammenhänge zwischen einzelnen Größen der kombinierten Raster ergeben. Ein eigener Versuch wurde immer komplexer, mit Brüchen, die auch bei kleiner Schreibweise mehr als ein DIN A 4 - Blatt quer füllen. Es handelte sich um den Versuch, wie oben durch Probieren ausgeführt, den Zusammenhang zwischen der gleichen Längung der beiden rautenförmigen Raster und der Weite des dazu möglichst passenden quadratischen Rasters so zu ermitteln, daß das primäre Moire möglichst klein wird. Nach geeigneter Zusammenfassung und Ersatz einiger komplizierter Glieder ergab sich sogar eine, so dachte ich, gute Formel. Als ich aber die Ergebnisse der Formel mit Hilfe des Programms betrachtete, stellte ich fest, daß dieser Zusammenhang zwar stimmt, daß aber die sekundären Moires sehr groß werden. Es ist einfach sehr schwierig, hier geeignete Zusammenhänge auszuwählen. Daher scheint es mir sinnvoller, mit dem Programm numerisch zu experimentieren. Insbesondere in kompilierter Form und auf einem schnellen Rechner könnten sich interessante Dinge ergeben. Das

oben stehende Ergebnis repräsentiert nur das mit meinen Mitteln erreichbare Optimum.

### 3.4. Anwendungen des Ergebnisses und des Programms

Eigentlich macht das Programm Voraussagen mit einem Modell für den Tiefdruck. Die technischen Möglichkeiten, Nöpfchen zu längen oder zu stauchen dürften begrenzt sein. Die Frage ist, ob diese Grenze bei der angewandten 1,5-fachen Längung liegt. Vielleicht wäre es möglich, sich mit einem Kompromiß an das Optimum anzulehnen. Vielleicht kann man auch Nöpfchen weiter längen, aber nicht stauchen. Insgesamt sähe ich hier die Möglichkeit, die Nöpfchen größer zu machen und damit die Gravurgeschwindigkeit zu erhöhen ohne Einbußen an der Druckqualität zu haben. Dies ist aber nur einem Hersteller von Gravurmaschinen möglich, somit für die Allgemeinheit von geringem Interesse.

Interessanter ist die Frage, ob das Ergebnis für die autotypischen Druckverfahren anwendbar ist, und ob es Vorteile bringt. Diese Art von Raster ist bestimmt anwendbar. Die Algorithmen die zur Umrechnung von Grauwerten in Rasterpunkte notwendig sind, wären sogar erheblich unkomplizierter als die bisherigen Algorithmen, die die photomechanische Rasterung nachahmen. Das Problem dabei ist, daß diese Rasteralgorithmen von den Scannern ausgeführt werden. Auf die zugehörige Software haben nur die Hersteller Einfluß. Hier selbst etwas umprogrammieren zu wollen, ist völlig aussichtslos. Wenn sich ein Scannerhersteller dafür erwärmen könnte, wäre das etwas anderes. Die andere Möglichkeit, ein Raster halbwegs professionell zu belichten, wäre über Laserbelichter. Hierbei wird heute zur Ansteuerung des Belichters in der Mehrzahl die Seitenbeschreibungssprache Postscript verwendet. Die einfachste Möglichkeit, ein Raster, wie das oben vorgeschlagene, zu verwirklichen, ist, rechteckige Rasterzellen anstelle quadratischer zu erzeugen. Bei Postscript ist dies grundsätzlich nicht möglich. Über einige Kniffe könnte man eine ähnliche Wirkung erreichen. Hier wäre allerdings ein Verhältnis der Rauten von 1 : 2 günstiger zu realisieren als ein Verhältnis von 1 : 2,5, wie es oben beschrieben ist. Soviel zur Realisierbarkeit.

Sollte dieses Raster direkt von einem Scannerhersteller angewendet werden, dann dürfte die Verarbeitungsgeschwindigkeit höher sein, da der nötige Rechenaufwand wesentlich geringer ist als bei herkömmlicher Rasterung. Die

Verdruckbarkeit eines Rasters hängt, wie oben schon besprochen, im wesentlichen von der Fläche ab, die pro Punkt zur Verfügung steht. Da die Grundform bei diesem Raster nicht quadratisch ist, muß man erst einige kleine Berechnungen anstellen. Die durchschnittliche Fläche einer Rasterzelle ergibt sich zu  $(1 \times 2,5 + 1 \times 2,5 + 1,2 \times 1,2) : 3 : 2 = 1,073$ . Diese Zahl ist die Verhältniszahl zu einem normalen Raster mit einer Fläche von 1. Dieses Raster hat bei einer normalen Winkelung eine primäre Moireweite von knapp 2 und eine sekundäre Moireweite von knapp 4, wie bereits oben besprochen. Wenn man nun die Moires des neuen Rasters entsprechend umrechnet, dann entsteht eine primäre Moireweite von 2,00 und eine sekundäre Moireweite von 2,82. Die entstehende Rosettenstruktur ist also weniger sichtbar als beim herkömmlichen Raster. Außerdem entstehen keine unendlichen Verwinkelungen, folglich kommt es zu keiner Farbdrift. Auch die Toleranzen des Rasters sind wesentlich größer. Es kann also "schlampiger" montiert werden und der Bogenlauf einer Druckmaschine hat ebenfalls höhere Toleranzen. Das wichtigste aber ist, daß auch der Filmtransport der Laserbelichter höhere Toleranzen haben darf, ohne daß es zu Moireerscheinungen kommt. Man kann dieses Raster auch noch weiter verbessern, indem man davon ausgeht, daß Schwarz keinen so hohen Tonwertumfang benötigt wie die beiden anderen "dunklen" Druckfarben Cyan und Magenta. Dann kann man nämlich die Fläche der beiden rautenförmigen Raster so definieren, daß sie der Fläche von 1 entsprechen, während das quadratische Raster, dem Schwarz zugeordnet ist, kleiner als eins ist. Dies dürfte die Qualität eines Farbsatzes nicht beeinträchtigen, da Schwarz normalerweise nur zur Unterstützung des Dunkelanteils der Buntfarben, als Skelettschwarz dient. In diesem Falle betrüge die primäre Moireweite 1,72 und die sekundäre Moireweite 2,424, was insgesamt schon wesentlich besser als beim herkömmlichen Raster ist. Als Nebeneffekt kommt hinzu, daß die Auflösung von Cyan und Magenta in unterschiedlichen Richtungen unterschiedlich groß ist, aber immer gegenläufig, so daß in Farben, in denen beide Grundfarben vorkommen, sich diese zu einer höheren Auflösung ergänzen können.

Ein Nachteil dieser Rasterungsart liegt in den unterschiedlichen Rasterweiten pro Farbe. Also müssen unterschiedliche Druckkennlinien berücksichtigt werden. Bei genügend Erfahrung mit einem solchen Raster dürfte dies aber keine größeren Probleme mehr machen. Umstellungen in den Vorstufen wären außer der anderen Belichterprogrammierung keine zu erwarten, anders als beim frequenzmodulierten Raster. Die unterschiedliche Rasterweite hätte bei Archivierung der Bilddaten sogar den Vorteil, daß man einen Teil redundanter

Daten, die für diese Rasterungsart eben nicht gebraucht werden, ohne Qualitätseinbußen weglassen könnte.

Insgesamt muß man aber viele Tests machen, bis man entscheiden kann, ob sich der Entwicklungsaufwand und die Programmierarbeit lohnen, ein solches Raster einzuführen. Abgesehen davon dürfte aber das größte Hindernis die Akzeptanz in der Praxis werden.

## 4. Überlegungen zur Gestaltung neuer Raster

### 4.1. Grenzen regelmäßiger Rasterung

Die Variation des Tiefdruckrasters hin zu einem allgemeinen autotypischen Raster, wie in 3.3. und 3.4. beschrieben, bringt schon einige Verbesserungen. Aber es dürfte schwer sein, unter eine Moireweite von 2 zu kommen. Bei Verkinkelungen der Raster durch Montage oder Papierverzug bleibt das Problem des Moires oder der Farbdrift, auch wenn die größten Schwierigkeiten etwas gemildert sind. Das Problem ist einfach nicht zu lösen, da man auf einem Winkel von  $180^\circ$  in der Fläche insgesamt mindestens sechs Linienstrukturen unterbringen muß, die sich nicht gegenseitig stören dürfen. Die sich bei oberflächlicher Betrachtungsweise ergebenden  $30^\circ$  sind einfach zu wenig.

Ein Ziel, speziell für den Flexodruck, wäre es außerdem, in den Lichtern oder den Tiefen Punkte wegzulassen, um damit eine geringere Linienzahl und damit einen größeren Tonwertumfang zu bekommen. Aus den vorhergehenden Ergebnissen sieht man leicht, daß ein solches Unterfangen bei einem regelmäßigen Raster plötzlich ganz neue Linienstrukturen ergibt. Als Folge davon treten völlig neue Moireerscheinungen auf. Da die Linienzahl kleiner wird, werden die zu erwartenden Moires größer und damit störender. Die Berechnungen solcher Raster dürften ferner sehr schwer fallen. In welche Richtung müssen Lösungen dieses Problems gehen?

### 4.2. Neue Lösungen

#### 4.2.1. Der Schritt von der Regelmäßigkeit zum Chaos

Wenn man das frequenzmodulierte Raster einmal außer acht läßt, das rechnerisch sehr schwierig zu verwirklichen ist, dann haben die herkömmlichen Raster Verschiedenes gemeinsam. Über eine konstante Frequenz wird eine Amplitude des Dunkelwertes moduliert. Problem dabei ist, daß die Grundfrequenz je nach Winkel anders ist. Durch Überlagerung der verschiedenen Frequenzen in verschiebter Richtung entstehen unerwünschte Muster. Außerdem ergänzt das Sehzentrum aus der Regelmäßigkeit bestimmte Linien. Es bleibt

also keine andere Möglichkeit, als die sich ergebenden Linienstrukturen zu verwischen. Wenn man nicht auf eine Frequenzmodulation hinaus will, dann bleibt eigentlich nur noch, auf die konstante Trägerfrequenz zu verzichten und statt dessen ein Rauschen zu benutzen. Aus in obigen Kapiteln beschriebenen Gründen soll aber über dieses Rauschen eine Amplitude moduliert werden. In die normale Ausdrucksweise übersetzt heißt das, ein Raster erzeugen, dessen Punktanordnung unregelmäßig ist, dessen Punktgröße aber in herkömmlicher Weise entsteht.

## 4.2.2. Wie chaotisch muß ein Raster sein?

Man könnte jetzt über das Ziel hinausschießen und meinen, je verrauschter, desto besser für ein Raster. Dazu muß man sich aber vor Augen halten, daß Computer, die die Raster letztendlich erzeugen, streng deterministische Maschinen sind, jedenfalls im Idealfall. Hier sei der kurze Hinweis auf das Buch von Joachim Graf "Murphys Computergesetze" gestattet. Der Computer hat zwar einen Zufallsgenerator, aber so einfach ist die Sache nicht. Die Punkte müssen nämlich nach wie vor am besten einheitliche Größenverhältnisse aufweisen, die nur über die Graustufe variieren. Außerdem sind die Graustufen, die in ein Raster verrechnet werden, immer in einem von der Speicherorganisation vorgegebenen, aus quadratischen Flächen bestehenden Datenfeld abgelegt. Von diesen Quadraten muß man ausgehen, da diese Art, ein Graustufenbild zu speichern, vorläufig die sinnvollste ist. Im frequenzmodulierten Raster wäre es eventuell sinnvoll, die Konturen mit ähnlichem Grauwert zu speichern, siehe dazu Punkt 1.3.2. . Es bleiben also quadratische Grundbausteine. Nun könnte man es sich sehr einfach machen. In einem vorgegebenen Rasterfeld einfach den Mittelpunkt des Rasterpunktes per Zufall innerhalb der quadratischen Rasterzelle verschieben. Der Rasterpunkt sollte dabei aber nicht an der Begrenzung der Zelle anstoßen. Dabei übersieht man leicht, daß dann die Anordnung der Rasterzellen sichtbar bleibt, nämlich durch die nicht mit Punkten besetzbaren weißen Ränder. Es würde aber zugegeben schon eine Verringerung der Kantenschärfe der Linien bewirken, deshalb ist diese Art zu rastern diejenige, die schließlich im Rahmen dieser Diplomarbeit noch verwirklicht werden konnte. Besser wäre es, man könnte die Grenzen der einzelnen Zellen gleichsam überschreiten. Diese Überschreitung ist insbesondere in Postscript schwer zu verwirklichen. Deshalb soll vorher genau überlegt werden, was man

eigentlich bewirken will. Es sollen Moire, Farbdrift und Rosette vermieden werden. Deshalb sollen folgende Forderungen gelten:

1. Durch Parallelverschiebung in gewissen Grenzen soll bei zwei Rastern keine nennenswerte Menge der Punkte genau zur Deckung gebracht werden können. Das heißt, bei Parallelverschiebung muß der gesamte Grauwert etwa konstant bleiben.
2. In dem Raster dürfen keine Muster irgendwelcher Art erkennbar sein. Folglich können sich auch kein Moire und keine Rosette ergeben.
3. Die Größe der Punkte gleichen Grauwertes sollte nicht zu sehr variieren.

Um Punkt eins erfüllen zu können müssen genügend verschiedene Anordnungen eines Punktes pro Grauwertquadrat möglich sein. Angenommen, man hat fünf Möglichkeiten. So ist die Chance, daß bei Überlagerung der gleich aufgebauten Raster der verschiedenen Farben zwei Punkte genau aufeinanderliegen,  $1 : 25$ , folglich decken sich nur 4% der Punkte völlig. Dies ist allerdings ein Beispiel, bei dem Verwinkelung der Raster und Größe der einzelnen Rasterpunkte nicht berücksichtigt wurden, es geht nur ums Prinzip.

Punkt zwei wird erfüllt, wenn nicht mehr als zwei Punkte einer einzigen wahrnehmbaren Struktur zugeordnet werden können. Diese Bedingung läßt sich im voraus schwer absichern. In einem späteren Beispiel wird sich zeigen, daß dafür am besten Tests geeignet sind. Als Struktur in diesem Sinn nimmt das Sehzentrum vor allem Linien wahr, aber auch Kurvenzüge machen sich störend bemerkbar. Strukturen können auch aus freien Stellen, also Nicht-Punkten bestehen, wie bereits weiter oben in diesem Kapitel festgestellt.

Punkt drei ist als Abgrenzung zum Kornraster zu verstehen, da dort durch Anhäufung von Feinpunkten verschiedene Punktgrößen entstehen können und so eine Gradationskurve schwer bestimmbar wird. Eine Variation der Punktfläche im Bereich von  $1 : 2$  dürfte noch keine größeren Auswirkungen haben, aber dies ist nur eine Vermutung und müßte durch Versuche näher untersucht werden.

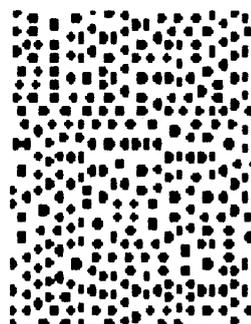
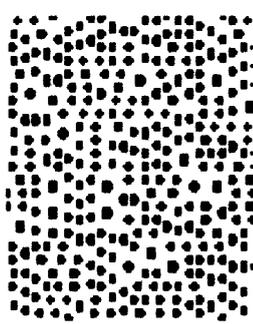
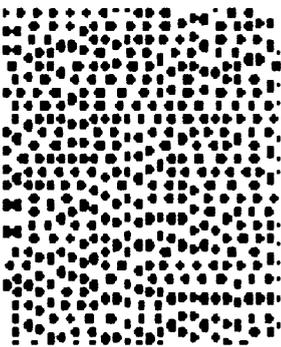
Insgesamt reichen also verhältnismäßig kleine Verwischungen, um der Wahrnehmung ein Rauschen vorzutäuschen. Mehr zu Verrauschen als nötig, macht nur die Ergebnisse unkontrollierbarer und steigert den Rechenaufwand. Das Kornraster schießt sozusagen über das Ziel hinaus, da drucktechnische Eigenschaften eines solchen Rasters nicht mehr oder nur schwer vorhersehbar sind.

### 4.2.3. Erster prinzipieller Vorschlag

### 4.2.3.1. Gedankenmodell

Da im letzten Absatz festgestellt wurde, daß man gar nicht so sehr viel Zufall in ein neues Raster einfließen lassen muß, schlage ich hier ein Rasterprinzip vor, das von sehr rationalen Größen ausgeht. Die Grundbausteine des Rasters sind Rechtecke. Zunächst gehen wir davon aus, daß in jedem Rechteck sich ein Punkt befindet. Um es nicht uferlos werden zu lassen habe ich folgende Rechtecke ausgewählt:  $1 : 1$ , also die normale Rasterzelle,  $1 : 2$ ,  $2 : 1$ , man muß hier unterscheiden, denn der Computer kann keine Ähnlichkeiten erkennen,  $1 : 3$ ,  $3 : 1$ ,  $2 : 2$ , siehe dazu auch die Skizze im nächsten Unterpunkt. Um eine Fläche zufällig, aber gleichmäßig mit diesen Formen auszufüllen habe ich mir ein Programm geschrieben, das auch gewissermaßen ein Modell anzeigt. Eines dieser Modelle habe ich sehr stark verkleinert einmal in verschiedenen Rasterkombinationen gedruckt. Da es sich aber um den ersten Versuch handelte, waren die Ergebnisse nicht gerade berauschend. Die Flächendeckung der einzelnen Punkte ist außerdem nicht konstant. Darauf habe ich an dieser Stelle noch keinen Wert gelegt. Man kann an den Modellen aber gut entstehende Strukturen wahrnehmen. Außerdem kann man, wenn man die Ausdrucke vor starkem Licht übereinanderhält, gut sehen, ob die Forderung nach der möglichst geringen gegenseitigen Überdeckung erfüllt ist.

Aufgrund dieser Erkenntnisse konnte man schon gewisse Entwicklungen machen, so erwiesen sich zum Beispiel das  $1 : 3$  und das  $3 : 1$  Rechteck als zu ungleichmäßig. Deshalb wurden diese Rechtecke mit jeweils zwei Punkten gefüllt, anstatt mit einem. Außerdem stellte sich heraus, daß das  $2 : 2$  Rechteck gegenüber den anderen Formen viel zu stark auffiel. An beiliegenden Skizzen



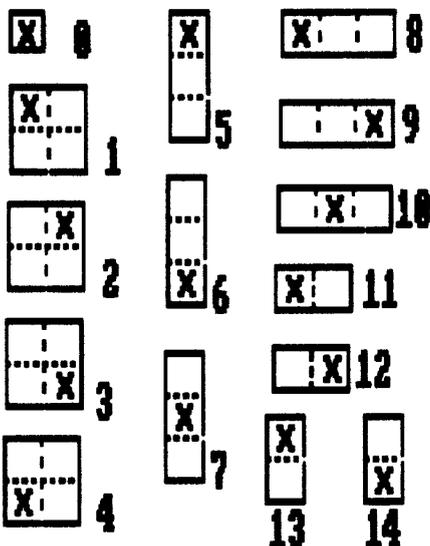
zen kann man die deutlichen Verbesserungen erkennen. Bei Überlagerung des so erzeugten Rasters entstanden trotzdem keine schlechteren

Eigenschaften als mit dem  $2 : 2$  Rechteck. An beiliegenden Skizzen kann man gut die Entwicklung sehen. Auch durch Gewichtung in der Häufigkeitsverteilung einzelner Punkte konnten Verbesserungen erzielt werden. Wenn man nämlich die Häufigkeitsverteilung nicht beeinflußt, werden mehr Grundquadrate als zum Beispiel  $1 : 3$  Rechtecke verwendet, da die größeren Formen oft nicht in

das Umfeld eingepaßt werden können und deshalb verworfen werden müssen.

### 4.2.3.2. Computerprogramm zur Erzeugung unregelmäßiger Rastermodelle

Das Programm füllt eine Fläche mit den im vorigen Punkt angegebenen Rechtecken vollständig aus. Dazu wird dem Programm ein Grundquadrat vorgeben und die neue Startposition. Das Programm wählt eine neue Form aus und stellt diese neben das Grundquadrat. Um diese Form wird nun das Umfeld kreisförmig geprüft. An der nächsten freien Position nach einer besetzten Position, die in diesem Fall vom Grundquadrat besetzt ist, legt das Programm die neue Startposition fest. Im Verlauf des Abarbeitens kommt es nun zu Fällen, an denen bei der Rundumprüfung mehrmals ein Übergang zwischen besetzter und unbesetzter Position erkannt wird, wobei die Bereichsgrenzen als besetzt gelten. Da das Programm nur zur letzten erkannten, unbesetzten Übergangposition springt, werden die anderen Übergangpositionen in einem Stapel abgelegt. Gibt es nun um eine Form keine freie Stelle mehr, wird die oberste abgelegte Übergangposition aus dem Stapel geholt. Das Programm arbeitet so lange, bis der Stapel abgearbeitet ist.



Wenn man von einer bestimmten Position aus zum Beispiel ein 1 : 3 Rechteck zeichnen will, ist auch noch die Frage, ob das aktuelle Grundquadrat im unteren, mittleren oder oberen Drittel des Rechtecks liegt. Diese Formen müssen deshalb noch einmal unterschieden werden. Durch solche Unterscheidungen entstehen insgesamt 14 Formen, wie sie in der Skizze dargestellt sind.

Das Programm ist wie das Programm oben in Amiga-Basic geschrieben. Eine Übertragung in einen anderen Basicdialekt oder

eine andere Programmiersprache dürften nicht all zu schwer fallen. Es handelt sich um eine Version, in der das 1 : 3 und das 3 : 1 Rechteck bereits mit jeweils 2 Punkten besetzt sind. Jedoch wird das 2 : 2 Rechteck mit verwendet. Das Verständnis dieses Programms dürfte etwas schwierig sein. Deshalb ist das Listing kommentiert. Aber auch mit diesem Kommentar braucht es einige Einarbeitung um alles zu verstehen. Aus technischen Gründen mußten einige Pro-

grammzeilen im Sinne von Basic auf mehrere Schreibzeilen verteilt werden, da sie zu lang waren. Sie sind besonders stark eingerückt.

CLS

Zuweisungen:

DEFINT a-Z

DIM x(200)

DIM y(200)

DIM Feld(6,6)

DIM Form(14,1,12)

DIM Nutzformen(18)

DIM formen(14)

DIM Pruef(12)

DIM a%(4)

DIM b%(4)

DIM xr(11)

DIM yr(11)

REM form0

DATA 1,0,-1,-1,-1,0,1,1,0,0,0,0,7

DATA -1,-1,-1,0,1,1,1,0,0,0,0,0,0

REM form1

DATA 2,1,0,-1,-1,-1,-1,0,1,2,2,2,11

DATA -1,-1,-1,-1,0,1,2,2,2,2,1,0,0

REM form2

DATA 1,0,-1,-2,-2,-2,-2,-1,0,1,1,1,11

DATA -1,-1,-1,-1,0,1,2,2,2,2,1,0,0

REM form3

DATA 1,0,-1,-2,-2,-2,-2,-1,0,1,1,1,11

DATA -2,-2,-2,-2,-1,0,1,1,1,1,0,-1,0

REM form4

DATA 2,1,0,-1,-1,-1,-1,0,1,2,2,2,11

DATA -2,-2,-2,-2,-1,0,1,1,1,1,0,-1,0

REM form5

DATA 1,0,-1,-1,-1,-1,-1,0,1,1,1,1,11

DATA -1,-1,-1,0,1,2,3,3,3,2,1,0,0

REM form6

Hier wird der Stapel für die Übergangspunkte definiert.

Hier werden die Daten für das jeweilige Umfeld der Formen angegeben, in relativen Koordinaten zum Startpunkt. Die erste Zeile beinhaltet jeweils die x-, die zweite jeweils die y-Koordinate eines Umfeldpunktes. Der letzte Wert der ersten Zeile gibt die Anzahl der möglichen Umfeldpunkte an.

```

DATA 1,0,-1,-1,-1,-1,-1,0,1,1,1,1,11
DATA -3,-3,-3,-2,-1,0,1,1,1,0,-1,-2,0
REM form7
DATA 1,0,-1,-1,-1,-1,-1,0,1,1,1,1,11
DATA -2,-2,-2,-1,0,1,2,2,2,1,0,-1,0
REM form8
DATA 3,2,1,0,-1,-1,-1,0,1,2,3,3,11
DATA -1,-1,-1,-1,-1,0,1,1,1,1,1,0,0
REM form9
DATA 1,0,-1,-2,-3,-3,-3,-2,-1,0,1,1,11
DATA -1,-1,-1,-1,-1,0,1,1,1,1,1,0,0
REM form10
DATA 2,1,0,-1,-2,-2,-2,-1,0,1,2,2,11
DATA -1,-1,-1,-1,-1,0,1,1,1,1,1,0,0
REM form11
DATA 2,1,0,-1,-1,-1,0,1,2,2,0,0,9
DATA -1,-1,-1,-1,0,1,1,1,1,0,0,0,0
REM form12
DATA 1,0,-1,-2,-2,-2,-1,0,1,1,0,0,9
DATA -1,-1,-1,-1,0,1,1,1,1,0,0,0,0
REM form13
DATA 1,0,-1,-1,-1,-1,0,1,1,1,0,0,9
DATA -1,-1,-1,0,1,2,2,2,1,0,0,0,0
REM form14
DATA 1,0,-1,-1,-1,-1,0,1,1,1,0,0,9
DATA -2,-2,-2,-1,0,1,1,1,0,-1,0,0,0
FOR n=0 TO 14
  FOR m=0 TO 1
    FOR l=0 TO 12
      READ Form(n,m,l)
    NEXT l
  NEXT m
NEXT n
xogrenze=600
yogrenze=180
xugrenze=0
yugrenze=0

```

Die Daten werden dem Feld  
Form(n,m,l) zugewiesen.

Die Grenzen der Graphik werden  
festgelegt.

```

x=50
y=50
GOSUB f13zeichn
x(1)=60
y(1)=55
Z=1
Hauptschleife:
  WHILE (Z>0)
    x=x(Z)
    y=y(Z)
    Z=Z-1
  REM Neue Form definieren
  n=0
  FOR xr=x-30 TO x+30 STEP 10
    m=0
    FOR yr=y-15 TO y+15 STEP 5
      IF (yr>yogrenze)OR(yr<yugrenze)OR(xr>xogrenze)OR(xr<xugrenze)THEN
        Feld(n,m)=-32767
      ELSE
        GET (xr,yr)-(xr,yr),a%
        GET ((xr+9),(yr+4))-((xr+9),(yr+4)),b%
        IF (a%(3)=0)AND(b%(3)=0) THEN
          Feld(n,m)=0
        ELSE
          Feld(n,m)=-32767
        END IF
      END IF
      m=m+1
    NEXT yr
    n=n+1
  NEXT xr
  FOR n=0 TO 14
    formen(n)=1
  NEXT n
  IF Feld(3,3)<>0 THEN Schleifenende
  IF Feld(4,3)<>0 THEN formen(1)=0:formen(4)=0: formen(8)=0: formen(11)=0

```

Der vorgegebene Startpunkt wird gezeichnet.

Startpunkt wird vom Stapel genommen.

Das Umfeld wird eingelesen. Vorsicht, die Daten werden mit GET direkt aus dem Bildschirmspeicher gelesen!

Alle Formen werden zunächst für gültig erklärt. Hier können auch Formen ausgeschlossen werden.

```

: formen(10)=0
IF Feld(5,3)<>0 THEN formen(8)=0
IF Feld(3,2)<>0 THEN formen(3)=0: formen(4)=0: formen(6)=0: formen(7)=0
: formen(14)=0
IF Feld(3,1)<>0 THEN formen(6)=0
IF Feld(4,2)<>0 THEN formen(4)=0
IF Feld(2,3)<>0 THEN formen(2)=0: formen(3)=0: formen(9)=0: formen(10)=0
: formen(12)=0
IF Feld(1,3)<>0 THEN formen(9)=0
IF Feld(2,2)<>0 THEN formen(3)=0
IF Feld(3,4)<>0 THEN formen(1)=0: formen(2)=0: formen(5)=0: formen(7)=0
: formen(13)=0
IF Feld(3,5)<>0 THEN formen(5)=0
IF Feld(2,4)<>0 THEN formen(2)=0
IF Feld(4,4)<>0 THEN formen(1)=0
Anzahlformen=0
FOR n=1 TO 14
  IF formen(n)=1 THEN
    Anzahlformen=Anzahlformen+1
    Nutzformen(Anzahlformen)=n
  END IF
NEXT n

```

Hier werden die Bedingungen einzeln geprüft, welche Formen zulässig sind und welche nicht.

Die zulässigen Formen werden in ein Feld geschrieben und sortiert.

Per Zufall wird eine Form gewählt.

```

Neueform=Nutzformen((INT(RND*(Anzahlformen+1))))
ON Neueform GOSUB f1zeichn,f2zeichn,f3zeichn,f4zeichn,f5zeichn,f6zeichn,
  f7zeichn,f8zeichn,f9zeichn,f10zeichn,f11zeichn,f12zeichn,f13zeichn,
  f14zeichn
IF Neueform=0 THEN GOSUB f0zeichn

```

REM Umfeld der neuen Form pruefen

```

Umfeldzahl=Form(Neueform,0,12)
Pruef(0)=Feld((3+Form(Neueform,0,Umfeldzahl)),(3+Form(Neueform,1,Umfeldzahl)))
FOR n=0 TO Umfeldzahl
  Pruef(n+1)=Feld((3+Form(Neueform,0,n)),(3+Form(Neueform,1,n)))
  IF (Pruef(n+1)=0 AND Pruef(n)<>0) THEN
    Z=Z+1
  
```

```

x(Z)=x+10*Form(Neueform,0,n)
y(Z)=y+5*Form(Neueform,1,n)
END IF
NEXT n

```

Die Übergangspunkte zwischen besetztem und unbesetztem Umfeld werden ermittelt und in den Stapel geschrieben.

Schleifenende:

```
WEND
```

f0zeichn:

```

PRESET (x,y)
LINE STEP (0,0)- STEP(9,4),,bf
LINE STEP(-6,-3)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
RETURN

```

Die einzelnen Punkte werden gezeichnet, hier die Form 0.

f1zeichn:

```

PRESET (x,y)
LINE STEP (0,0)- STEP(19,9),,bf
CIRCLE STEP(-10,-5),3,2
CIRCLE STEP(1,0),3,2
CIRCLE STEP(0,1),3,2
CIRCLE STEP(-1,0),3,2
PAINT STEP(0,0),2
RETURN

```

Hier wird die Form 1 gezeichnet, und so weiter.

f2zeichn:

```

PRESET (x-10,y)
LINE STEP (0,0)- STEP(19,9),,bf
CIRCLE STEP(-10,-5),3,2
CIRCLE STEP(1,0),3,2
CIRCLE STEP(0,1),3,2
CIRCLE STEP(-1,0),3,2
PAINT STEP(0,0),2
RETURN

```

f3zeichn:

```
PRESET (x-10,y-5)
LINE STEP (0,0)- STEP(19,9),,bf
CIRCLE STEP(-10,-5),3,2
CIRCLE STEP(1,0),3,2
CIRCLE STEP(0,1),3,2
CIRCLE STEP(-1,0),3,2
PAINT STEP(0,0),2
RETURN
```

f4zeichn:

```
PRESET (x,y-5)
LINE STEP (0,0)- STEP(19,9),,bf
CIRCLE STEP(-10,-5),3,2
CIRCLE STEP(1,0),3,2
CIRCLE STEP(0,1),3,2
CIRCLE STEP(-1,0),3,2
PAINT STEP(0,0),2
RETURN
```

f5zeichn:

```
PRESET (x,y)
LINE STEP (0,0)- STEP(9,14),,bf
LINE STEP(-6,-4)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
LINE STEP(1,-8)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
RETURN
```

f6zeichn:

```
PRESET (x,y-10)
LINE STEP (0,0)- STEP(9,14),,bf
LINE STEP(-6,-4)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
LINE STEP(1,-8)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
RETURN
```

f7zeichn:

```
PRESET (x,y-5)
LINE STEP (0,0)- STEP(9,14),,bf
LINE STEP(-6,-4)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
LINE STEP(1,-8)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
RETURN
```

f8zeichn:

```
PRESET (x,y)
LINE STEP (0,0)- STEP(29,4),,bf
LINE STEP(-8,-3)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
LINE STEP(-14,-1)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
RETURN
```

f9zeichn:

```
PRESET (x-20,y)
LINE STEP (0,0)- STEP(29,4),,bf
LINE STEP(-8,-3)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
LINE STEP(-14,-1)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
RETURN
```

f10zeichn:

```
PRESET (x-10,y)
LINE STEP (0,0)- STEP(29,4),,bf
LINE STEP(-8,-3)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
LINE STEP(-14,-1)-STEP(3,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-5,0),2
RETURN
```

f11zeichn:

```
PRESET (x,y)
LINE STEP (0,0)- STEP(19,4),,bf
LINE STEP(-12,-3)-STEP(5,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-7,0),2
RETURN
```

f12zeichn:

```
PRESET (x-10,y)
LINE STEP (0,0)- STEP(19,4),,bf
LINE STEP(-12,-3)-STEP(5,2),2,bf
PRESET STEP(1,-1),2: PRESET STEP(-7,0),2
RETURN
```

f13zeichn:

```
PRESET (x,y)
LINE STEP (0,0)- STEP(9,9),,bf
LINE STEP(-6,-6)-STEP(3,3),2,bf
LINE STEP(-4,-2)-STEP(5,1),2,bf
RETURN
```

f14zeichn:

```
PRESET (x,y-5)
LINE STEP (0,0)- STEP(9,9),,bf
LINE STEP(-6,-6)-STEP(3,3),2,bf
LINE STEP(-4,-2)-STEP(5,1),2,bf
RETURN
```

Dieses Programm erzeugt nur Modelle. Es ist noch kein Programm, mit dem man im eigentlichen Sinn ein Raster erzeugen kann. Wie die Schritte von diesem Programm zu einem anwendbaren Raster aussehen, wird im nächsten Kapitel beschrieben. Leider ist ein solches Raster in Postscript nur mit Abstrichen zu realisieren, weshalb noch zwei weitere Vorschläge folgen sollen. Eine Routine zum Ausdrucken ist nicht eingebaut. Die Ausdrücke wurden realisiert, indem das Bild mit einem Hintergrundprogramm in einen externen Speicher gelesen und schließlich mit einem anderen Programm ausgedruckt wurde. Es handelt sich also lediglich um eine Art indirekte Hardcopy des Bildschirms. Das Programm gibt also nur eine ungefähre Vorstellung. Selbst mit einem

Nadeldrucker wären noch wesentlich bessere Ergebnisse möglich.

### 4.2.3.3. Realisierung des Rastermodells

Um so ein Raster zu realisieren, ist es natürlich möglich, die Graustufen zu lesen und selbst die einzelnen Rasterpunkte aus den Grundpunkten des Belichters zusammensetzen. Das hat den Nachteil, daß ein Bild möglichst in einer vorgegebenen Auflösung abgetastet werden muß, um in der Ausgabe ein Bild mit bestimmter Größe zu erhalten. Alles muß außerdem auf das Ausgabegerät abgestimmt werden. Um von Ein- und Ausgabegeräten weitgehend unabhängig zu sein und um in weiten Grenzen skalieren zu können, wurde die Seitenbeschreibungssprache Postscript geschaffen. Da der Postscripttreiber, in diesem Fall spricht man allerdings schon von einem Interpreter, da Postscript sehr leistungsfähig und schon optimal an das Ausgabegerät angepaßt ist, muß man sich hier keine weiteren Gedanken über irgendwelche Hardware machen.

Für die weitere Beschreibung ist es deshalb sinnvoll, die mathematischen Modelle, die Postscript benutzt, zu kennen. Da der Halbtonmechanismus üblicherweise nicht umprogrammiert wird, finden sich nur in wenigen Postscript-handbüchern genauere Beschreibungen. Hier sei auf "Das große Buch zu Postscript" von Tobias Weltner verwiesen. Da ein konkretes Programm erst in einem späteren Beispiel folgen soll, genügt hier eine kurze Beschreibung des mathematischen Modells der Spotfunktion. Die Rasterweite ist zunächst einmal nicht interessant und Winkelung benötigt dieses Raster keine.

Da die Auflösung des Ausgabegerätes nicht bekannt ist, wird jede einzelne Rasterzelle durch ein Koordinatensystem beschrieben mit  $x$  Element  $[-1;+1]$  und  $y$  Element  $[-1;+1]$ . Diese beiden Variablen dienen als Eingangsdaten. In der Realisierung in Postscript liegen sie als oberste Werte auf dem Stack, wenn die Spotfunktion aufgerufen wird. Nun kann vom Anwender eine zweiparametrische Funktion  $f(x,y)$  definiert werden, die der jeweiligen Koordinate  $(x,y)$  eine Priorität zuordnet. Die Ergebnisse müssen ebenfalls zwischen  $-1$  und  $+1$  liegen. Der Interpreter prüft dann zunächst welche Matrix er zur Verfügung hat. Bei einem Laserbelichter sind dies üblicherweise  $16 \times 16$ , also 256 realisierbare Graustufen. Es kann aber zum Beispiel auch eine Matrix von  $5 \times 5$  sein. Jedem Punkt dieser Matrix wird nun mit Hilfe der Spotfunktion eine Priorität zugeordnet. Ist nun die Matrix am Belichter zu füllen, prüft der Belichter zuerst wieviele der Punkte gesetzt werden müssen. Die Punkte werden nun ihrer Priorität nach gesetzt, zum Beispiel kreisförmig oder schachbrettförmig. Da die Postscriptfunktion mit dem vollen Sprachumfang von Postscript pro-

grammiert werden kann, kann die Spotfunktion auch je nach Grauwert anders sein. Interessantestes Beispiel bisher ist die Übertragung des FOGRA Rasterpunktmodells auf Postscript. Siehe dazu den Bericht vom FOGRA Symposium vom 17. und 18. April in München, den Bericht von Dr. Karl Haller und Wolfram Fischer.

Die Spotfunktion eignet sich also hervorragend, um Punktformen zu variieren. Aber für das Raster braucht man Punkte, die über die Grenzen der einzelnen Rasterzellen hinweggehen. Wie erreicht man das? Dies soll am Beispiel eines Punktes erklärt werden, der aus zwei Rasterzellen nebeneinander besteht. Man definiert bei der linken Rasterzelle einfach einen Rasterpunkt, der seine höchste Priorität an der Rechten Zellenkante in der Mitte hat. Analog dazu definiert man in der rechten Rasterzelle einen Punkt, der seine höchste Priorität an der linken Kante der Rasterzelle hat. Da die Punkte an der trennenden Kante beider Rasterzellen immer zusammentreffen, werden sie vom Auge als ein einziger Punkt wahrgenommen. Ein guter Nebeneffekt dabei ist, daß die Auflösung des Grundrasters nicht verlorenggeht, da der Punkt eigentlich aus zwei verschiedenen Grauwerten besteht. Für jedes Rechteck müssen also alle Grundrasterzellen definiert werden. Also zwei Grundrasterzellen für das Rechteck 2 : 1, zwei andere Grundrasterzellen für das Rechteck 1 : 2 und so weiter. Insgesamt benötigt man 11 Grundrasterzellen, wenn man das Rechteck 2 : 2 nicht verwendet.

Das Problem ist nun, dem Postscriptinterpreter zu sagen, welche Grundrasterzelle er gerade verwenden muß. Dazu muß man zunächst eine Matrix definieren, in der die Nummer der jeweiligen Grundrasterzelle steht. Diese Matrix kann man zum Beispiel mit dem Programm im vorigen Unterpunkt erzeugen, wenn man es noch etwas ändert. Der Interpreter muß dann so programmiert werden, daß er jeweils eine Zeile der Matrix abarbeitet. Ist die Bildzeile noch nicht zu Ende, benützt er noch einmal diese Zeile, bis die Bildzeile voll ist. Dann springt er zur nächsten Zeile der Matrix. Wenn alle Zeilen der Matrix abgearbeitet sind, fängt er wieder von vorne an. Die Matrix wird also vertikal und horizontal so lange aneinander gesetzt, bis das Bild gefüllt ist. Damit kein Moire entsteht, muß man die Matrix nur etwas verschieben, so daß auch bei schlampiger Montage die Matrix in keinem Falle mit sich selbst genau überlagert wird. Das erreicht man, indem man bei weitem Farben eben nicht beim Element (0,0) anfängt, sondern bei der zweiten Farbe beim Element (20,20), bei der dritten Farbe beim Element (40,40) und bei der vierten Farbe beim Element (60,60). Der Abstand sollte mindestens 2 mm betragen. Also muß

bei höherer Auflösung auch die Matrix größer sein. Angenommen, der Belichter arbeitet mit einer Rasterweite von 60 Linien /cm, dann muß die Matrix mindestens 48 x 48 Grundrasterzellen repräsentieren, mehr wäre aber besser.

Die Matrix könnte theoretisch genauso groß wie oder größer als das Bild sein. Außerdem könnte die Matrix für das Motiv jedesmal neu maßgeschneidert werden, um den Konturen gut folgen zu können. Hier wäre noch einiges an Entwicklung möglich. Durch die unregelmäßige Anordnung der Punkte könnte auch das verwirklicht werden, was bei regelmäßigen Rastern eben nicht geht, nämlich in den Lichtern weniger Punkte zu schreiben. Dazu soll später noch eine ausführliche Idee beschrieben werden.

Warum läßt sich aber das im vorigen Absatz vorgestellte Raster so nicht verwirklichen? Bei den Rechtecken 1 : 3 und 3 : 1 liegen die Punkte nämlich nicht auf der Kante zwischen den Rasterzellen. Bei sehr geringen Tonwerten gibt es keine Verbindung zwischen den einzelnen Bestandteilen der Rasterpunkte in den verschiedenen Rasterpunktzellen mehr. Ohne diese beiden Rechteckarten ist aber die Variationsbreite zu gering. Deshalb ist ein anderer Vorschlag notwendig.

## 4.2.4 Versuch in Postscript

### 4.2.4.1 Grundsätzliche Überlegungen

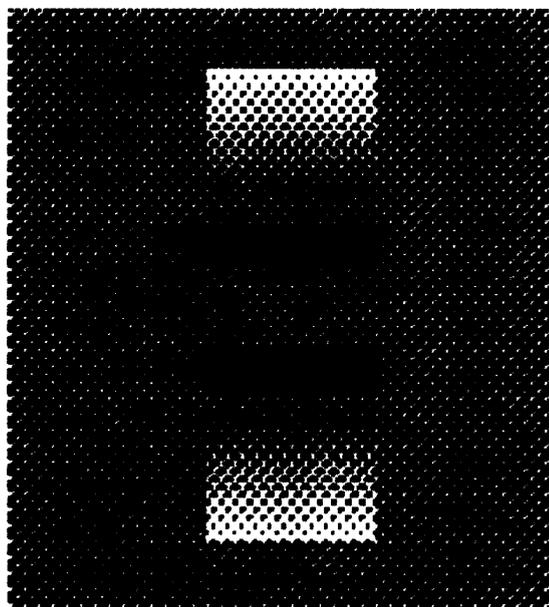
Der Vorschlag in 4.2.3 scheiterte letztendlich an den Grenzen von Postscript. Nun soll hier also der Versuch gemacht werden, den Sprachumfang von Postscript auszunutzen und möglichst viel von den im theoretischen Teil erarbeiteten Verbesserungen zu implementieren. Dazu möchte ich auf einen Vorschlag zurückgreifen, der in Punkt 4.2.2 schon einmal angeklungen ist, dort aber verworfen wurde. Wie wäre es also, wenn man von einer quadratischen Grundzelle ausgeht, aber den Mittelpunkt des Rasterpunktes innerhalb dieses Grundquadrates zufällig verschiebt? In diesem Fall müßte man aber nach wie vor winkeln. Man hätte aber die Rosettenstruktur weitgehend aufgelöst. Dabei ergibt sich aber das große Problem, daß schwarze Punkte und Nicht-Punkte in feste Beziehungen treten müssen. Wenn man also nur jeweils eine Rasterzelle berücksichtigt, dann ergeben sich drucktechnisch äußerst nachteilige Formen für die Nicht-Punkte oder weißen Punkte. Man muß also viele Punkte gemeinsam betrachten und miteinander verknüpfen.

## 4.2.4.2 Zur Programmieretechnik

Wie kann man mehrere Punkte miteinander verknüpfen? In Punkt 4.2.3.3 wurde die theoretische Grundlage der setscreen-Funktion beschrieben. Diese Funktion kann nun aber nicht nur ein geschlossener mathematischer Ausdruck sein, sondern eine beliebige Funktion im Sinne einer Programmiersprache. Die Überlegung ist nun die, eine Funktion zu definieren, die innerhalb einer Rasterzelle nicht einfach einen Punkt erzeugt, sondern mehrere. Für meine weiteren Überlegungen und im praktischen Beispiel gehe ich von einer Punktanordnung innerhalb der Rasterzelle von 10 x 10 Punkten aus, also insgesamt 100 Punkte. Man denkt sich nun innerhalb dieser Rasterzelle weitere Rasterzellen in einer Anordnung von ebenfalls 10 x 10. In jeder dieser Fein-Rasterzellen sitzt nun ein Punkt, dessen Mittelpunkt innerhalb dieser Fein-Rasterzelle per Zufall festgelegt wird. Dabei wurden zwei Regeln zusätzlich eingeführt, erstens muß der Mittelpunkt 0,2 vom Rand der Feinzelle entfernt sein, wenn man deren Kantenlänge als 1 annimmt, zweitens müssen der x-Wert und der y-Wert des Feinpunktes um 0,2 von den x- beziehungsweise y-Werten verschieden sein, damit keine Reihen entstehen. Die definierte setscreen-Funktion sollte zudem möglichst einfach sein. Je komplizierter die setscreen-Funktion, desto länger dauert die Belichtung eines Bildes, die mit einem Postscript-RIP sowieso nicht gerade den Ruf besonderer Schnelligkeit hat. Also ist es sinnvoll, möglichst viele Werte schon in einer Tabelle abzulegen und Postscript zur Verfügung zu stellen. Hierzu habe ich pro Feinzelle 5 x 5 Stützpunkte gewählt. Zur Erzeugung einer Datei, die die Daten an Postscript übergibt, habe ich ein Programm in Basic geschrieben, das schließlich einen ASCII-Code erzeugt. Da Postscript ebenfalls als ASCII-Datei abgespeichert ist, kann man die Zahlen bereits so abspeichern, daß sie von Postscript als Feld-Definition akzeptiert werden. Dazu muß man vor die Zahlenkolonne nur die Worte `"/Variablenname ["` und nach der Zahlenkolonne `"] def"` anhängen. Wenn man nun diese Datei mit einer bestehenden Postscriptdatei zusammenhängt, dann erkennt Postscript dieses Feld. Pro Zelle mit je 50 Stützpunkten, also 5 Stützpunkte pro Punkt, habe ich je ein Feld gebildet.

Die Größe der Rasterzellen der setscreen-Funktion ist völlig unabhängig von den angenommenen Grauwertzellen eines Bildes. Das bedeutet für diesen Fall, daß trotz der zehn mal so groben Rasterzelle der setscreen-Funktion die feineren Grauzellen eines Bildes wiedergegeben werden. Dies klingt im ersten Moment ziemlich unverständlich. Dazu muß man sich die Funktionsweisen der

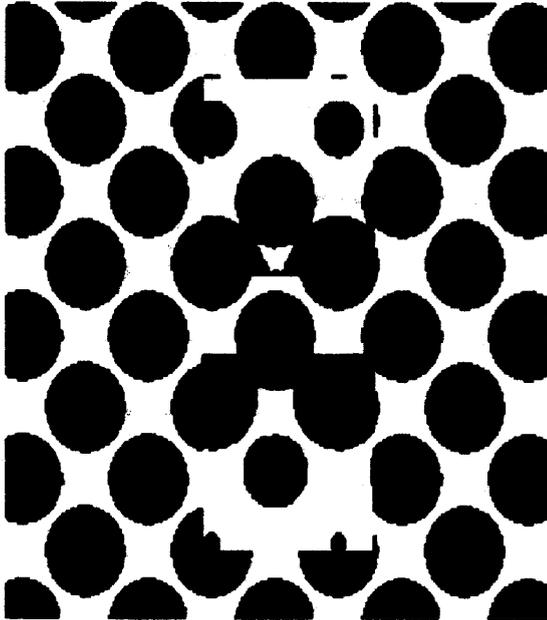
setscreen- und der image-Funktion vor Augen halten. Die image-Funktion ermittelt zunächst einen Grauwert und die Fläche auf der dieser Grauwert herrschen muß. Dann ermittelt die image-Funktion, wie viele Belichter-, Laserdrucker- oder auch Nadeldruckerpunkte für dieses Graufeld zur Verfügung stehen. Sie ermittelt also ein Rechteck, im Normalfall ein Quadrat, von  $m \times n$  Punkten. Dieses Rechteck hat nichts mit der Rasterzelle der setscreen-Funktion zu tun. Der Übergang erfolgt erst jetzt. Zur Erinnerung: Die Rasterzelle der setscreen-Funktion ist in Koordinaten eingeteilt  $x$  von  $-1$  bis  $+1$  und  $y$  von  $-1$  bis  $+1$ . Nun wird jeder  $n,m$ -Druckerpunkt genommen und es wird geprüft, welche  $x$ -,  $y$ -Koordinaten er in der aktuellen Rasterzelle der setscreen-Funktion hat. Die  $x$ - und die  $y$ -Koordinate werden an die setscreen-Funktion übergeben. Diese gibt dann einen Prioritätswert zwischen  $-1$  und  $+1$ , an. Schließlich werden die Punkte der  $n,m$ -Druckerzelle mit der Rasterzelle (!) der Priorität nach besetzt. Wenn zum Beispiel ein Grauwert von 50% erzeugt werden soll, dann werden die 50% der Punkte mit der höchsten Priorität besetzt. Alle anderen Punkte werden nicht gesetzt. Für die image-Funktion ist es also völlig egal, in welcher Rasterzelle der setscreen-Funktion sie gerade Punkte setzt, entscheidend ist nur, welche  $x$ - und welche  $y$ -Koordinate dieser Punkt hat. Ferner ändert sich der Grauwert irgendwo innerhalb eines Bildes. Diese Änderung kann mitten durch einen Rasterpunkt



verlaufen, was dann zu sehr merkwürdigen Rasterpunktformen führen kann. Dies kann man sehr leicht feststellen, indem man ein Bild mit ausgeprägten Konturen in einem sehr groben Raster in Postscript ausgibt. Zur Illustration sind einige Bilder beigefügt, die von einem Postscriptinterpreter erzeugt wurden. Das erste Bild zeigt das Bild in einem normalen Raster, wie es für einen Nadeldrucker geeignet ist. Zum Vergleich dazu ein sehr grob gerastertes Bild. Man kann dabei gut die Verzerrungen

der einzelnen Punkte sehen. Aus dem ursprünglichen Bild ist aber eindeutig zu erkennen, daß die einzelne Grauzelle wesentlich kleiner ist als ein Rasterpunkt. Zur weiteren Verdeutlichung ist noch eine ähnliche Vorlage mit dem sehr groben Raster im Anhang an dieses Kapitel beigefügt, an der die Effekte

noch etwas deutlicher sichtbar sind. Als Folgerung kann man festhalten, daß es egal ist, ob die Rasterzelle im vorgeschlagenen Raster zehn mal so grob ist, wenn sie dafür innerhalb der Rasterzelle wieder zehn Punkte erzeugt, ergeben



sich keine Qualitätsverluste.

Vorhin sprach ich von fünf Stützstellen pro Feinpunkt. Dies wäre natürlich zu grob. Deshalb muß zwischen den Stützpunkten interpoliert werden. Dabei gehe ich davon aus, daß der gegebene  $x,y$ -Punkt in einem Quadrat zwischen vier Stützpunkten liegt. Da man zwischen vier Punkten nicht eindeutig linear interpolieren kann, muß man ein Dreieck wählen. Dazu wird der Mittelpunkt zwischen den Punkten bestimmt und es wird ihm eine Priorität vom Durchschnitt der vier Prioritäten zugeordnet. Nun müssen das Dreieck und der Punkt in ihrer Lage normiert werden. Ich habe eine Normierung gewählt, bei der das entstehende gleichschenklige Dreieck mit dem  $90^\circ$ -Winkel nach rechts zeigt. Diese Normierung ist notwendig, damit die nachfolgende Formel gilt. Zunächst numeriert man mit der linken, oberen Ecke des Quadrates beginnend die Ecken von Punkt1 bis Punkt4 durch. Man ordnet nun dem obersten Eckpunkt des Dreiecks den Buchstaben C zu, der  $90^\circ$ -Ecke A und dem unteren Eckpunkt B. Die  $x$ - und  $y$ -Koordinaten bleiben. Und die Priorität eines Punktes wird als  $p(P)$  bezeichnet. Die Normierung geschieht folgendermaßen: wenn  $x > y$  ist, dann ist  $p(B)$  gleich  $p(\text{Punkt2})$ , gleichzeitig werden  $x$  und  $y$  vertauscht; im anderen Falle ist  $p(B)$  gleich  $p(\text{Punkt4})$ ; wenn  $(x+y) > 1$  ist, dann ist  $p(C)$  gleich  $p(\text{Punkt3})$ , gleichzeitig erhält  $y$  den Wert von  $1-x$  und  $x$  den Wert von  $1-y$ ; im anderen Falle ist  $p(C)$  gleich  $p(\text{Punkt1})$ . Die Formel für die Interpolation lautet dann  $p(x,y) = p(A)(2x) + p(B)(y-x) + p(C)(1-x-y)$ .

Man mag sich nun fragen, ob der Aufwand nicht etwas übertrieben ist. Mit der angegebenen Formel reicht aber ein Datenbestand von fünf Stützstellen pro Feinpunkt aus. Das ergibt bei einer Rasterzelle von  $10 \times 10$  Feinpunkten immerhin schon einen Datenbestand von  $50 \times 50$ , also 2500 Werten. Dies ist durchaus von Bedeutung, da zumindest in meinem Postscript-Interpreter das Zuordnen von Feldvariablen relativ langsam ging. Außerdem kann man mit dieser Formel beliebig fein interpolieren. Das bedeutet für den Anwender, man

bleibt hundertprozentig geräteunabhängig. In einer ersten Programmversion versuchte ich, die Berechnungen der Punkte alle im Postscript selbst zu erledigen. Außerdem waren die Probleme mit der Interpolation und der Anordnung der Nicht-Punkte nicht gelöst. Ich versuchte mich mit Rundungen und anderen Vereinfachungen durchzuhangeln. Die Ergebnisse waren niederschmetternd. Leider brauchte es dann wieder viel Kraft, um noch einmal alles umzuwerfen und neu anzufangen. Ohne eine saubere mathematische Lösung ist eben nichts zu machen. Das Postscriptprogramm in seiner jetzigen Form tut also nichts besonderes. Es ermittelt abhängig von  $x$  und  $y$  die vier nächsten Stützpunkte und deren Prioritäten. Es normiert und interpoliert in der angegebenen Weise und stellt den so berechneten Prioritätswert auf dem Stack bereit.

### 4.2.4.3 Das Postscriptprogramm

Im eigentlichen Sinne handelt es sich hier nicht um ein vollständiges Postscriptprogramm. Dieses Programm erzeugt nur sechs Graufelder auf einer DIN A4-Seite, in verschiedenen Graustufen. Die linken und die rechten drei Graufelder haben dieselben Tonwerte, sind aber anders gewinkelt. Die linken Graufelder sind mit  $45^\circ$  gewinkelt, die rechten mit  $0^\circ$ . Dieses Programm soll lediglich die setscreen-Funktion in einer Anwendung zeigen. Die alten setscreen-Werte werden in diesem Programm nicht gesichert, also vorsicht mit der Anwendung! Wenn man das Programm in seiner jetzigen Form anwendet, ist das Ausgabegerät danach in einem anderen Zustand. In diesem Fall würde nur ein reset helfen. Das Programm ist etwas unübersichtlich. Auch die beigefügten Kommentare dürften daran nichts Wesentliches ändern. Es ergab sich nämlich das Problem der Geschwindigkeit. Also habe ich versucht, möglichst wenig Variablendefinitionen zu verwenden, sondern mit dem Stack zu arbeiten. Die Variablen sind als Integer-Variable gespeichert und werden erst in der Funktion durch 100 geteilt. Auch das spart Rechengeschwindigkeit. Eine weitere Schwierigkeit wird daher rühren, daß ich das Feld, das die Funktionen enthält, zunächst nicht zuordnen lasse, indem ich die Klammern um die Feldnamen gesetzt habe, zum Beispiel (F0). Auf diese Weise wird verhindert, daß der Interpreter gleich beim Aufruf des Feldes "Felder" sämtliche 2500 Werte auf den Stack legt. Erst wenn entschieden ist, welche Felder tatsächlich gebraucht werden, werden diese mit den Befehlen "cvx cvn" vollständig entschlüsselt und auf den Stack gelegt. Dies schont erheblich den Speicherplatzbedarf des Stacks. Bei den verhältnismäßig kleinen Feldern macht dies noch nicht so viel aus. Wenn man aber mehr und feinere Punkte pro Rasterzelle

haben will, fällt dieser Faktor massiv ins Gewicht. Bei Verständnisschwierigkeiten empfehle ich das Buch von Tobias Weltner, "Das große Buch zu PostScript", das seinem Namen wirklich alle Ehre macht. Ich halte es in Sachen Übersichtlichkeit und Geschlossenheit auch für wesentlich besser als die Bücher von Adobe. Man sollte bei diesem Buch schon über Programmierkenntnisse verfügen.

```
/F0 [51 50 12 -16 25 54 43 -23 -44 5 81 61 -28 -15 56 51 9 -28
-4 46 71 62 -13 -52 12 68 64 33 7 44 83 80 41 26 66 91 70 34 -27 2
42 68 80 -38 -20 61 77 54 -2 25 51] def
```

```
/F1 [67 71 30 -3 20 33 6 -52 -81 -56 52 47 -24 27 84 64 30 -3 22
59 81 79 24 -25 37 88 71 43 8 33 57 42 -10 -18 45 76 63 29 -25 8 46
64 74 -48 -33 45 66 44 -8 29 67] def
```

.  
 . %Hier würden nur weitere Zahlenkolonnen folgen, die das Vorprogramm erzeugt.  
 .

```
/F50 [51 50 12 -16 25 54 43 -23 -44 5 81 61 -28 -15 56 51 9 -28
-4 46 71 62 -13 -52 12 68 64 33 7 44 83 80 41 26 66 91 70 34 -27 2
42 68 80 -38 -20 61 77 54 -2 25 51] def
```

```
/Felder [ (F0) (F1) (F2) (F3) (F4) (F5) (F6) (F7) (F8)
(F9) (F10) (F11) (F12) (F13) (F14) (F15) (F16) (F17)
(F18) (F19) (F20) (F21) (F22) (F23) (F24) (F25) (F26)
(F27) (F28) (F29) (F30) (F31) (F32) (F33) (F34) (F35)
(F36) (F37) (F38) (F39) (F40) (F41) (F42) (F43) (F44)
(F45) (F46) (F47) (F48) (F49) (F50) ] def
```

```
/feinraster 50 def
```

```
/feingrob{                                     %Die x- und y-Koordinaten wer-
                                               %den in Adressen für die Felder
1 add 2 div dup                               %und in Feinkoordinaten zer-
                                               %den in Adressen für die Felder
50 mul cvi                                    %und in Feinkoordinaten zer-
```

---

```
/zwischen exch def                                %legt und auf den Stack gelegt

zwischen 50 div

sub

zwischen

} bind def

/Neuspot{                                          %Hier beginnt die eigentliche

feingrob                                          %Funktion

/ygrob exch def                                  %x- und y-Wert werden der

/yfein exch def                                  %Funktion Feingrob übergeben

feingrob                                          %und Variablen zugeordnet

/xgrob exch def

/xfein exch def

Felder dup ygrob get                            %die Prioritätswerte der vier

cvx cvn load dup xgrob get                      %umgebenden Stützpunkte werden

100 div                                          %vom Stack geholt und durch

/Punkt1 exch def                                 %100 geteilt

xgrob 1 add get 100 div

/Punkt2 exch def
```

---

```
ygrob 1 add get cvx cvn load dup

xgrob get 100 div

/Punkt4 exch def

xgrob 1 add get 100 div

dup /Punkt3 exch def

Punkt1 add Punkt2 add           %Der Durchschnitt der Prioritäten

Punkt4 add 4 div                 %wird gebildet

/PvonA exch def

xfein yfein gt {                %Es wird entsprechend der oben

  /PvonB Punkt2 def              %beschriebenen Bedingungen

  xfein yfein                    %normiert, damit die angegebene

  /xfein exch def                %Formel zur Interpolation ange-

  /yfein exch def                %wandt werden kann

} {

  /PvonB Punkt4 def

} ifelse

xfein yfein add 1 gt {

  /PvonC Punkt3 def

  1 xfein sub 1 yfein sub
```

---

```
/xfein exch def

/ymfein exch def

} {

/PvonC Punkt1 def

} ifelse

PvonA 2 mul xfein mul           %die Formel, wie oben angegeben,

PvonB ymfein xfein sub mul add  %in Postscript programmiert

1 xfein sub ymfein sub

PvonC mul add

} bind def

%Neue Screen wird gesetzt

currentscreen

pop                               %Die alte Spotfunktion und die

exch pop                          %alte Rasterweite werden verwor-

2 cvi exch %neue Rasterweite Linien/Inch

{ Neuspot } %neue Spotfunktion    %fen, die neuen gesetzt

setscreen

/SetAngle {                       %Diese Unterfunktion setzt eine
```

---

```
currentscreen                                %neue Winkelung. Sie ist dem
3 1 roll                                    %oben erwähnten Buch auf Seite
pop                                          %314 entnommen.
3 1 roll
setscreen
} bind def
45 SetAngle
/cm {72 mul 2.54 div} bind def              %Es flogen einfache Postscript-
/viereck {                                  %Anweisungen zum Zeichnen der
    moveto                                  %Quadrate.
    5 cm 0 cm rlineto
    0 cm 5 cm rlineto
    -5 cm 0 cm rlineto
    closepath
} bind def
newpath 3 cm 3 cm viereck
.10 setgray fill
newpath 3 cm 10 cm viereck
.5 setgray fill
```

```
newpath 3 cm 17 cm viereck
```

```
.95 setgray fill
```

```
0 SetAngle
```

```
newpath 10 cm 3 cm viereck
```

```
.10 setgray fill
```

```
newpath 10 cm 10 cm viereck
```

```
.5 setgray fill
```

```
newpath 10 cm 17 cm viereck
```

```
.95 setgray fill
```

```
showpage
```

```
% Ende des Programmtextes
```

#### 4.2.4.4 Zum Vorprogramm

Das vorhige Kapitel beschreibt nur die Funktionsweise der eigentlichen set-screen-Funktion. Das Programm ist aber in gewisser Weise "dumm", es verarbeitet nur die Stützstellen-Punkte. Nun braucht man aber, wie unter 4.2.4.1 schon beschrieben ein Programm, das diese Stützpunkte erzeugt. Einige Kriterien für die Anordnung sind schon unter 4.2.1.1 angegeben. Zur Programmierung habe ich mich wieder der Sprache Basic bedient, da es hier mehr auf Verständlichkeit als auf Geschwindigkeit ankommt. Mit einem RND-Befehl werden die Punkte gesetzt. Wenn die erwähnten Kriterien nicht zutreffen, werden die Punkte verworfen und neu berechnet. Nun kommt der eigentliche Clou. Wohin kommen die Nicht-Punkte? Sie grundsätzlich auf die Feinzellen-Kreuzung fallen zu lassen, wäre die einfachste Möglichkeit, aber damit herrscht wieder das Rosetten-Problem, wie schon mehrfach erwähnt, nur eben im negativen Bereich. Außerdem führt dies zu sehr unregelmäßigen Punktfor-

men. Am konsequentesten ist es, sich zunächst vier Punkte, die im "Quadrat" zueinander stehen zu betrachten. Sie stehen natürlich nicht im Quadrat, weil sie in ihrer Feinzelle unregelmäßig angeordnet sind. Nun legt man je eine Gerade zwischen den zwei Punkten, die sich diagonal gegenüberstehen. Am Schnittpunkt beider Geraden legt man den Nicht-Punkt fest. Wenn man die Punkte links oben beginnend im Uhrzeigersinn durchnumeriert, dann ergeben sich für den Nicht-Punkt folgende Formeln zur Berechnung seiner Koordinaten:

$$X = [ (x_2-x_3)(y_1x_4-y_4x_1) + (x_1-x_4)(y_3x_2-y_2x_3) ] : [ (x_2-x_3)(y_1-y_4) + (x_1-x_4)(y_3-y_2) ]$$

$$Y = y_1 + [ (y_4-y_1)(X-x_1) ] : (x_4-x_1)$$

Da die Punkte in dem Programm nie direkt untereinander oder nebeneinander liegen können, müssen keine Einschränkungen für die Werte gemacht werden. Man erhält in dem Programm zunächst also zwei Datenfelder, eines mit den Punkten und eines mit den Nicht-Punkten. Dazu sei noch bemerkt, daß die Felder etwas größer sind, da sie die Randkoordinaten hinten beziehungsweise unten noch einmal enthalten, um besonderen Regeln für die Randabfrage entgegen zu können.

Nun hat man aber erst die Orte, an denen Punkte und Nicht-Punkte liegen. Wie bekommt man aber nun die einzelnen Werte der Stützstellen? Dazu macht man es ähnlich wie die image-Funktion im Postscript-Interpreter. Es werden einfach x- und y-Koordinaten mittels einer FOR-NEXT-Schleife vorgegeben. Von diesem Punkte aus werden der nächste Punkt und der nächste Nicht-Punkt gesucht. Zur Erinnerung sei noch einmal daran erinnert, daß die höchste Priorität 1 und die niedrigste 0 ist. Den Abstand des aktuellen Punktes zum nächsten Punkt bezeichnen wir mit "a", den Abstand zum nächsten Nicht-Punkt mit "b". Ist also der aktuelle Punkt gleich weit vom nächsten Punkt und vom nächsten Nicht-Punkt entfernt, also  $a=b$ , dann erhält er die Priorität 0. Liegt er näher am nächsten Punkt, also  $a < b$ , dann gilt:  $\text{Priorität} = 1 - (a:b)$ . Liegt der aktuelle Punkt näher am nächsten Nicht-Punkt, also  $a > b$ , dann gilt:  $\text{Priorität} = -[ 1 - (b:a) ]$ . Man könnte nun die Priorität noch einer Funktion unterwerfen. Was sich hier besonders eignet, müßte erst noch ausgetestet werden. Der Einfachheit halber habe ich mich entschieden, es ersteinmal bei diesem linearen Zusammenhang zu belassen. Der Wert wird noch mit 100 multipliziert und gerundet, da zwei Stellen als Genauigkeit genügen. Auf diese Weise befinden sich in dem erzeugten Postscript-Feld nur Integer-Variablen, was eine etwas höhere Verarbeitungsgeschwindigkeit zur Folge hat. Erst vor der weiteren Verarbeitung werden die Werte vom Postscriptprogramm

wieder durch 100 geteilt. Die zusätzlichen Zeichenketten sind Postscript-Konventionen zur Variablendefinition, wie bereits unter 4.2.4.1 beschrieben.

### 4.2.4.5. Das Vorprogramm

Das Programm wurde im vorigen Kapitel schon weitgehend besprochen. Da es in Basic geschrieben ist, dürfte es keine größeren Verständnisschwierigkeiten geben. Ich habe das Programm aber mit einigen Bemerkungen versehen. Sie stehen in der jeweils rechten Spalte. Sie dürfen natürlich so nicht in einem lauffähigen Programm stehen. Vorsicht bei einigen überlangen Zeilen. Sie wurden einfach in der nächsten Zeile fortgesetzt, obwohl es in Basic eigentlich eine logische Zeile sein müßte.

```
CLS
DEFINT a-z
DEFSNG xab, yab, abstand, prioritaet, x, y
DIM x(12,12)                Felddefinitionen
DIM y(12,12)
DIM xn(12,12)
DIM yn(12,12)
DIM abstand(9)
DIM Ausgabe(50,50)
OPEN "Postzufall" FOR OUTPUT AS #1    Die Datei wird geöffnet.
a$="###"

FOR m=1 TO 10                Die Punkte werden per Zufall und
  FOR n=1 TO 10              nach bestimmten Regeln gesetzt.
    Rechnung:
      x(n,m)=RND/10*.6+.02+.1*(n-1)
      y(n,m)=RND/10*.6+.02+.1*(m-1)
      IF m=1 THEN x(n,11)=x(n,m): y(n,11)=y(n,m)+1
      IF m=10 THEN x(n,0)=x(n,m): y(n,0)=y(n,m)-1
      IF n=1 THEN x(11,m)=x(n,m)+1: y(11,m)=y(n,m)
      IF n=10 THEN x(0,m)=x(n,m)-1: y(0,m)=y(n,m)
      IF (ABS(x(n,m)-x(n,m-1))<.02) OR (ABS(y(n,m)-y(n-1,m))<.02)
        THEN Rechnung
    NEXT n
  NEXT m
x(0,0)=x(10,10)-1: y(0,0)=y(10,10)-1    Da das Feld überlappend ist,
```

```
x(11,0)=x(1,10)+1: y(11,0)=y(1,10)-1
x(11,11)=x(1,1)+1: y(11,11)=y(1,1)+1
x(0,11)=x(10,1)-1: y(0,11)=y(10,1)+1
```

müssen noch die Eckpunkte gesetzt werden.

```
FOR m=0 TO 10
  FOR n=0 TO 10
```

Hier werden die Positionen der Nicht-Punkte gesetzt und in ein Feld geschrieben.

```
  x1=x(n,m)
```

```
  x2=x(n+1,m)
```

```
  x3=x(n,m+1)
```

```
  x4=x(n+1,m+1)
```

```
  y1=y(n,m)
```

```
  y2=y(n+1,m)
```

```
  y3=y(n,m+1)
```

```
  y4=y(n+1,m+1)
```

```
  xn(n,m)=((x2-x3)*(y1*x4-y4*x1)+(x1-x4)*(y3*x2-y2*x3))/((x2-x3)*(y1-y4)+(x1-x4)*(y3-y2))
```

```
  yn(n,m)=y1+(y4-y1)*(xn(n,m)-x1)/(x4-x1)
```

```
  NEXT n
```

```
NEXT m
```

```
FOR m=0 TO 49
```

Die ersten Zeichen werden nach Postscript-Vereinbarungen in die Datei geschrieben.

```
  z$="Feld"+STR$(m)+" ["
```

```
  PRINT #1,z$;
```

```
  FOR n=0 TO 49
```

```
    spalte=INT(n/5)+1
```

```
    zeile=INT(m/5)+1
```

```
    xfein=1/50*n
```

```
    yfein=1/50*m
```

```
    links=spalte-1
```

```
    oben=zeile-1
```

```
    rechts=spalte+1
```

```
    unten=zeile+1
```

```
    xab=x(spalte,zeile)-xfein
```

```
    yab=y(spalte,zeile)-yfein
```

```
    abstand(0)=SQR(xab^2+yab^2)
```

Es wird jeweils die Entfernung zu einem Umgebungspunkt errechnet.

```
xab=xfein-x(rechts,oben)
yab=y(rechts,oben)-yfein
abstand(1)=SQR(xab^2+yab^2)
```

Diese Berechnung wird für die nächsten 9 Umgebungspunkte ausgeführt.

```
xab=xfein-x(rechts,zeile)
yab=yfein-y(rechts,zeile)
abstand(2)=SQR(xab^2+yab^2)
```

```
xab=xfein-x(rechts,unten)
yab=yfein-y(rechts,unten)
abstand(3)=SQR(xab^2+yab^2)
```

```
xab=xfein-x(spalte,unten)
yab=yfein-y(spalte,unten)
abstand(4)=SQR(xab^2+yab^2)
```

```
xab=xfein-x(links,unten)
yab=yfein-y(links,unten)
abstand(5)=SQR(xab^2+yab^2)
```

```
xab=xfein-x(links,zeile)
yab=yfein-y(links,zeile)
abstand(6)=SQR(xab^2+yab^2)
```

```
xab=x(links,oben)-xfein
yab=y(links,oben)-yfein
abstand(7)=SQR(xab^2+yab^2)
```

```
xab=x(spalte,oben)-xfein
yab=y(spalte,oben)-yfein
abstand(8)=SQR(xab^2+yab^2)
```

```
abstandp=abstand(0)
FOR l=1 TO 8
  IF abstand(l)<abstandp THEN abstandp=abstand(l)
NEXT l
```

Es wird überprüft, welcher der 9 Abstände der kürzeste ist.

```

spalte=INT((n/5)+.5)
zeile=INT((m/5)+.5)
rechts=spalte+1
unten=zeile+1

xab=xn(spalte,zeile)-xfein
yab=yn(spalte,zeile)-yfein
abstand(0)=SQR(xab^2+yab^2)

```

```

xab=xfein-xn(rechts,zeile)
yab=yfein-yn(rechts,zeile)
abstand(1)=SQR(xab^2+yab^2)

```

```

xab=xfein-xn(rechts,unten)
yab=yfein-yn(rechts,unten)
abstand(2)=SQR(xab^2+yab^2)

```

```

xab=xfein-xn(spalte,unten)
yab=yfein-yn(spalte,unten)
abstand(3)=SQR(xab^2+yab^2)

```

```

abstandn=abstand(0)

```

```

FOR l=1 TO 3

```

```

  IF abstand(l)<abstandn THEN abstandn=abstand(l)

```

```

NEXT l

```

```

IF abstandn=abstandp THEN prioritaaet=0

```

```

IF abstandn>abstandp THEN prioritaaet=(abstandp/abstandn)-1

```

```

IF abstandp>abstandn THEN prioritaaet=1-(abstandn/abstandp)

```

```

Ausgabe(n,m)=INT(prioritaaet*100)

```

```

IF Ausgabe(n,m)>=0 THEN PRINT "88"; ELSE PRINT "..";

```

```

PRINT #1,USING a$;Ausgabe(n,m);

```

```

NEXT n

```

```

PRINT #1,USING a$;Ausgabe(0,m);

```

```

PRINT #1,"] def"

```

```

PRINT " "

```

```

NEXT m

```

```

PRINT #1,"Feld50 ["

```

Nun erfolgt analog die Prüfung der Nicht-Punkte, allerdings werden nur 4 Umgebungspunkte überprüft. Mehr sind aus logischen Gründen nicht nötig.

Der kürzeste Abstand zu einem Nicht-Punkt wird ermittelt.

Nachdem die Prioritäten, wie im vorigen Kapitel beschrieben, errechnet wurden, werden die Werte mit 100 multipliziert, die Nachkommastellen werden verworfen und nach Postscript-Verein-

```

FOR n=0 TO 50
  PRINT #1,USING a$;Ausgabe(n,0);
NEXT n
PRINT #1,") def"

CLOSE

REM Ende des Programms

```

barung in die Datei gespeichert. Dabei werden Werte größer 0 als "88" angezeigt. Die anderen Werte werden als ".." angezeigt, damit man schon am Bildschirm ein grobes Bild bekommt.

#### 4.2.4.6 Weitere Vorteile des Rasters

Hauptzweck des Rasters war es, das Problem der Rosette mindestens zu mildern und das Raster unempfindlicher gegen Störungen in der Winkelung, wie schlechte Montage, Filmverzug und Papierlauffehler, zu machen. Nun hat dieses Programm aber noch einige höchst interessante Nebeneffekte. Da die Punkte nämlich unregelmäßig angeordnet sind, ist auch der Punktschluß über einen weiten Bereich verteilt. Es gibt also keinen plötzlichen Anstieg der Tonwertzunahme, sondern die Tonwertzunahme steigt langsam an. Einen ähnlichen Effekt gibt es auch beim Tonabriß. Hier passiert es nicht einfach so, daß bei einer Graustufe noch Punkte zu sehen sind und bei der anderen nicht mehr, wie es bis jetzt der Fall ist. Da die Punkte unregelmäßig angeordnet sind, haben sie auch eine etwas abweichende Ausdehnung, das heißt, während einige Punkte schon so spitz sind, daß sie nicht mehr ausdrucken, sind andere noch so groß, daß man sie noch sieht. Das bedeutet, die Punkte werden im Druckbild allmählich weniger, bis schließlich keine mehr vorhanden sind. Dabei handelt es sich in den besonders hellen und den besonders tiefen Tönen, wo ein ähnlicher Effekt auftritt, um eine Art frequenzmodulierter Raster. Da es sich hierbei um Effekte handelt, die je nach Druckverfahren variieren, müssen erst noch genaue Versuche gemacht werden, um das Verhalten des Rasters in den extremen Tönen im einzelnen zu untersuchen. Das hätte jedoch den Rahmen dieser Diplomarbeit bei weitem gesprengt. Wenn also die theoretischen Überlegungen alle zutreffen und wenn man annimmt, die Entwicklungen zur Optimierung dieses Rasters werden vollzogen, so dürften damit die wesentlichen Probleme der Vierfarbrasterung entscheidend gemildert sein. Wenn man sich nun noch vor Augen hält, daß dies in der angegebenen Form auf jedem postscriptfähigen Belichter realisierbar ist, dürfte dieses Raster auch in wirtschaftlicher Hinsicht interessant sein.

Zum besseren Verständnis wurde auch hier eine Vorlage am Ende des

Kapitels mit abgebildet. Das Motiv ist schon aus Punkt 4.2.4.2. bekannt, nur ist es hier in dem neuen Raster ausgeführt. Man erkennt deutlich die einzelnen Effekte des Rasters. Ebenfalls deutlich sichtbar ist die Wiederholung in 10 x 10er Quadraten. Hier wird klar, daß man hier noch optimieren muß. Auch ganze Bilder sind in dieser Rasterungstechnik möglich.

Wie überall hat aber alles seinen Preis. In diesem Fall heißt das, die einzelnen Farbauszüge enthalten im Punktbereich leichte Tonwertabweichungen. Wenn man also zum Beispiel einen glatten 50%-Ton erzeugen will, so wird er, wenn man ihn mit dem Densitometer ausmißt, nur statistisch um diesen Wert schwanken. Man kann also diese Art der Rasterung auch so auffassen:

Es werden bereits bei der Repro leichte Tonwertschwankungen gezielt eingebaut. Diese mindern dann aber die unkontrollierbaren Tonwertschwankungen, die bei der Überlagerung der vier Farben eintreten.

Deshalb ist diese Art von Raster auch niemals bei Schwarz-Weiß-Bildern zu empfehlen. Sie wird bei ihnen immer zu einem schlechteren Ergebnis führen als ein herkömmliches Raster. Wie stark sich allerdings die gezielten Tonwertschwankungen bemerkbar machen, muß erst noch durch Druckversuche ermittelt werden. Wahrscheinlich lassen sie sich so reduzieren, daß sie im Verhältnis zu den Vorteilen vernachlässigt werden können.

Abbildung eines groben Postscriptrasters  
nach Punkt 4.2.4.2.

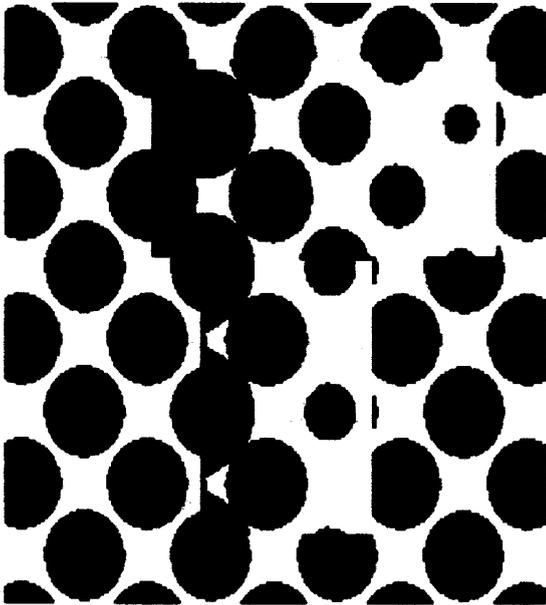
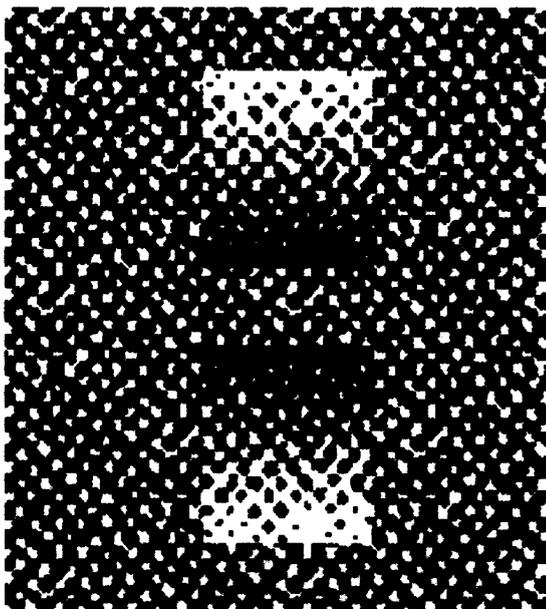


Abbildung eines statistisch verzerrten  
Rasters nach Punkt 4.2.4.6.



## 5. Zur Praxis des Belichtens von Rastern

### 5.1. Arbeiten an Postscriptsystemen

Wenn ein Postscriptprogramm syntaktisch richtig ist müßte es auch auf einem Postscriptbelichter ausgegeben werden, sollte man meinen. Daß dem nicht ganz so ist, wird man erfahren, wenn man Programme hat, die den Sprachumfang etwas weiter als üblich ausnutzen. Für die Versuche arbeitete ich hauptsächlich an einem 486er MS-DOS-Rechner, der über eine Parallelschnittstelle mit einem Page Image Processor (PIP) der Firma Autologic verbunden war. Dieser wiederum konnte wahlweise einen Laserdrucker oder einen Laserbelichter ansteuern. Vorversuche machte ich immer zu Hause an meinem Amiga mit Hilfe eines Shareware-Programms, das Postscriptfiles auf dem Bildschirm oder einem Nadeldrucker sichtbar machen kann.

Zu Hause konnte ich meine Postscriptfiles ohne Probleme abarbeiten lassen. Die Qualität auf einem Bildschirm oder auf einem Nadeldrucker ist zwar schlecht, aber man kann schon einiges erkennen. Mit Hilfe des Programms Photo-Styler konnte ich auch ohne Probleme Daten für Vierfarbsätze erzeugen lassen. Es ist auch kein Problem mit Hilfe eines normalen ASCII-Editors diese Daten in ein bestehendes Programm einzubinden. Bei den ersten Versuchen am PIP tat sich aber erst einmal gar nichts. Bei der Untersuchung von Postscriptfiles, die von Programmen erzeugt waren, entdeckte ich ziemlich kompliziert aufgebaute Header und Footer. Diese müssen sehr kurz sein, deshalb werden mittels Definitionen die Befehle dermaßen verkürzt, daß ein Nachvollziehen des Programmes nicht mehr möglich ist, was aber auch daher rühren kann, daß diese Programme oft sehr tief in die Speicherverwaltung und Datenübertragung einsteigen. Um dies zu umgehen, "lieh" ich mir Header und Footer von einem Programmgenerierten File aus. Es tat sich zwar schon etwas, aber es gab noch nicht die gewünschten Ergebnisse. Bei einem Test zu Hause wurden die Dateien jedoch problemlos ausgegeben. Der Grund für das Versagen war so nicht herauszubekommen. Postscript gibt zwar standartisierte Fehlermeldungen aus. Diese werden an dem erwähnten PIP aber nur durch Anschließen eines weiteren PCs über die serielle Schnittstelle ausgegeben. Dieses PIP hat einen eigenen Bildschirm, der im normalen Betrieb fast nicht ausgenutzt ist. Da müßte es doch möglich sein, Fehlermeldungen über diesen auszugeben. Aber leider

war es nicht der Fall.

Die Daten wurden also auf MacIntosh-Format übertragen, was relativ einfach möglich war, da es sich im Prinzip nur um ASCII-Code handelt. Bei den ersten Versuchen am Laserdrucker kamen nur viele unklare Fehlermeldungen. Schließlich war eine der vielen "Verzweiflungstaten" die, den Befehl "showpage" an eine andere Stelle des Programms zu schreiben. Plötzlich kam sehr schnell ein völlig richtiger Druck aus dem Laserdrucker. Syntaktisch sind beide Stellen zulässig. Deshalb machte der heimische Computer auch keine Probleme. Mit der neuen Datei schließlich noch ein Versuch am ursprünglichen PIP, und auch hier eine tadellose Ausgabe am Laserdrucker in etwa 50% der Fälle, je nachdem ob die Datenübertragung Funktionierte.

Nun wäre es logisch anzunehmen, daß wenn der Druck auf dem Laserdrucker funktioniert, er über denselben PIP auch auf dem Belichter funktioniert. Aber leider gibt es auch hier Fußangeln. Zuerst wurde mehrmals die Schwarzform an den PIP geschickt, leider ohne Erfolg. Also doch wieder der Weg zum MacIntosh, und dem Linotype-RIP (Raster Image Processor). Dort wurde auch antstandslos belichtet. Aber leider waren bei diesem RIP die Rasterweiten bestimmten Vorgaben unterworfen. Meine Rasterpunktform wurde zwar belichtet, da diese aber 10 mal so grob sein muß wie üblich, da mit jedem Punkt eine 10 x 10 Matrix ausgegeben wird (siehe Punkt 4.2.4.2.), war das Ergebnis niederschmetternd. Es wurde ein 10 mal so feiner Raster ausgegeben. Der neue RIP, der beliebige Rasterweiten und Winkel akzeptiert, sollte erst am nächsten Tag geliefert werden.

Also wieder zurück zum ursprünglichen System zu einem letzten Versuch. Diesmal wurde die Cyanform an den Belichter geschickt. Diese wurde plötzlich problemlos akzeptiert. Das Ergebnis war einwandfrei. Also wurden alle Farbauszüge an den Belichter geschickt. Gelb funktionierte ebenfalls ohne Probleme. Magenta wurde auch noch belichtet. Die Schwarzform war aber auch nach vielen Versuchen mit Neustarten des Systems und ähnlichem nicht zu bekommen. Also wurden alle Filme noch einmal genau untersucht und siehe da, in der Magentaform war bei einem Bild die vorgegebene Winkelung vom PIP einfach durch 0° ersetzt worden. Bei weiteren Belichtungsversuchen, war ein Magentafilm nicht mehr zu erhalten. Einige Versuche mit neuen Headern und ähnlichem schlugen ebenfalls fehl. Was hatten nun Schwarz und Magenta gemeinsam? Der Fehler mußte offensichtlich in der Winkelung liegen. Cyan war mit 45° gewinkelt, Gelb mit 0°, Schwarz und Magenta hatte ich beide rational gewinkelt, also Magenta mit  $75,963757^\circ$  ( $\arctan 4$ ) und Schwarz mit  $14,036243^\circ$

( $\arctan 0.25$ ). Aus Erfahrungen am eigenen Computer weiß ich, daß so gewinkelte Raster mehr Speicherplatz benötigen als Raster mit  $0^\circ$  oder  $45^\circ$ . Deshalb habe ich mich auch für eine rationale Rasterung entschieden, da diese wesentlich weniger Speicherplatz benötigt als zum Beispiel eine Winkelung von genau  $15^\circ$  beziehungsweise  $75^\circ$ .

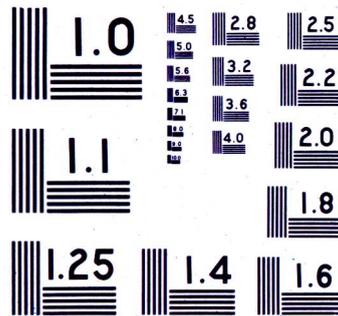
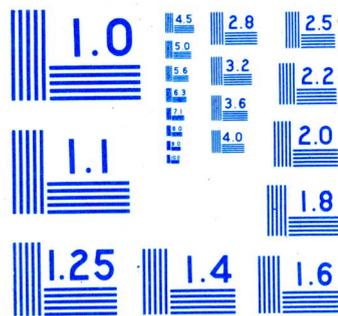
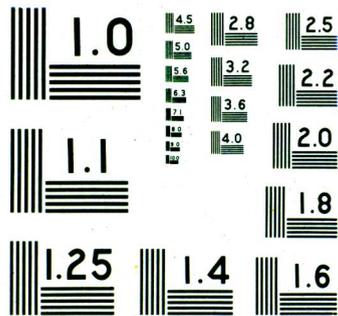
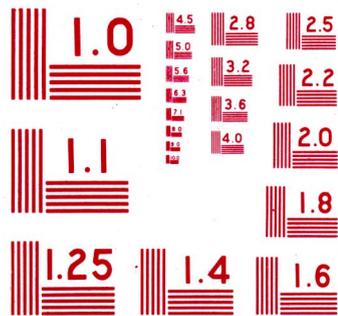
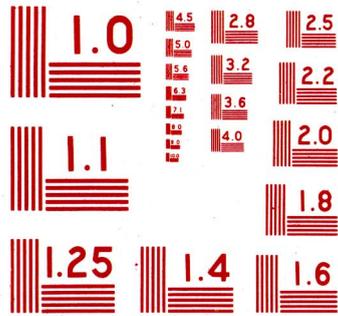
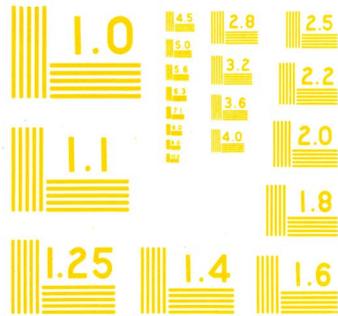
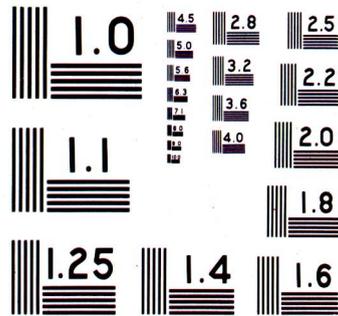
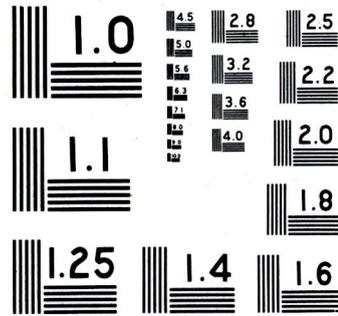
## 5.2. Auswertung der Ergebnisse

Unter Auswertung sollte man sich hier nicht zu viel vorstellen. Letztendlich geht es nur um eine subjektive Bewertung der Filme. Da aber Druckergebnisse letztendlich immer von einem Subjekt bewertet werden, ist eine solche nicht gerade wissenschaftliche Betrachtung in diesem Falle wohl zulässig.

Obwohl zunächst noch nicht alle Filme vorhanden sind, kann man durch Übereinanderlegen der Filme schon einige Tendenzen erkennen. Die einzelne Form sieht im herkömmlichen Raster besser aus als in meinem statistisch verzerrten Raster. Wenn man aber mehrere Farben überlagert, dann treten beim herkömmlichen Raster die üblichen Moires auf. Diese kommen bei dem neuen Raster nicht vor. Der herkömmliche Raster weist auch eine feinere Zeichnung auf. Bei Überlagerung mehrerer Filme erreicht die Zeichnung des neuen Rasters fast die Güte des herkömmlichen. Bei dem neuen Raster treten störende Effekte auf, wenn die Belichterauflösung nicht hoch genug ist. Zum Vergleich befanden sich auf der Testform nämlich ein 30er und ein 60er Raster. Im 30er Raster traten die störenden Effekte, die das neue Raster an und für sich hat, weniger in Erscheinung als beim 60er Raster.

Ein interessanter Vergleich läßt sich mit einer Neuentwicklung anstellen. Wenn man nämlich die Ergebnisse einer Heidelberger GTO-DI mit dem Fadenzähler betrachtet, dann sehen sie ähnlich aus wie das statistisch verzerrte Raster. Nach eigenen Aussagen verwendet Heidelberg ein normales Postscript-Raster und erreicht die Verzerrung erst durch physikalische Vorgänge beim Schreiben der Druckplatte in der Maschine. Bei dieser Maschine ist jedenfalls die Qualität, die mit einem 30er Raster erzeugt wird so gut, daß ein erster flüchtiger Blick leicht von zum Beispiel einem 48er Raster ausgehen kann. Nach eigener Aussage ist Heidelberg nicht in der Lage ein ähnlich gutes Raster auf Film zu erzeugen. Vielleicht ist es auch nur eine Hemmschwelle, daß der einzelne Film dann wesentlich schlechter aussähe als bei einem üblichen Raster. Erst bei einem Übereinanderdruck werden die deutlichen Vorteile sichtbar.

# Testdruck für Experiment nach Punkt 2.2.2.



# Vierfarbdruck mit statistisch verzerrtem Raster nach Punkt 5.2.



---

## Literaturverzeichnis

1. Bestenreiner, Friedrich: Vom Punkt zum Bild: Entwicklung, Stand und Zukunftsaspekte der Bildtechnik / Karlsruhe 1988
2. Bestenreiner, Friedrich: Verfahren zur Überlagerung von Punktrastern / Deutsche Patentschrift 1975, 25 20 646 6
3. Fischer, Gerhard: Der frequenzmodulierte Bildaufbau - ein Beitrag zum Optimieren der Druckqualität / Institutsbericht Darmstadt 1986
4. Fischer, Gerhard: Verfahren zur Herstellung von Druckformen mittels unregelmäßig verteilter Druckpunkte / Europäische Patentschrift 1981 0 074 422 B1
5. Fischer, Wolfram: Erzeugung von optimierten Rasterpunktformen mit der PostScript-Spotfunktion / fogra Symposium München 1991
6. Haller, Karl Dr.: Erzeugung von optimierten Rasterpunktformen mit der PostScript-Spotfunktion / fogra Symposium München 1991
7. Häuser, Frank: Die Entstehung des Farbeindrucks bei der autotypischen Farbmischung / Dissertation Bonn 1982
8. Morgenstern, Dieter: Rasterungstechnik fotomechanisch und elektronisch / Frankfurt am Main 1985
9. Neugebauer, Hans E. J.: Zur Theorie des Mehrfarbenbuchdruckes / Reprint Leipzig 1989
10. Scheuter, Karl R. Prof. Dr.: Verfahren zur Herstellung von Druckformen mittels unregelmäßig verteilter Druckpunkte / Europäische Patentschrift 1981, 0 074 422 B1
11. Schnitzler, Thomas Martin: Wo sind die Grenzen der Rasterfeinheit bei der Farbproduktion für den Offsetdruck? / fogra-Mitteilungen Nr. 141/142 Seite 10 ff
12. Schoppmeyer, Johannes Dr.: Zur Überdeckung der Rasterpunkte und Farbtenschwankungen im Mehrfarbendruck, Ergänzungen zur Theorie des Mehrfarbenrasterdrucks / Der Polygraph 17-82 Seite 1510 ff
13. Schoppmeyer, Johannes Dr.: Das "OPTI"-Rastersystem, Ein neues Rasterungssystem zur Verringerung des Moires und der Farbtenschwankungen beim Mehrfarbendruck / Der Polygraph 8-83 Seite 496 ff
14. Weltner, Tobias: Das große Buch zu PostScript / Düsseldorf 1991

Da streng nach Autoren geordnet ist, sind manche Beiträge, die von zwei Autoren stammen, doppelt aufgeführt.