

**NKF-Library für die computerunterstützte  
Suche von trainierten Objekten  
in Realzeit**

## Einführung

Die Funktionen der NKF-Bibliothek ermöglichen das Wiederfinden von einmal gelernten Bildausschnitten im Bildspeicher der PCgrab. Die gelernten Bildausschnitte werden im folgenden als *Masken* bezeichnet. Ein Beispiel zeigt Bild 1 und Bild 2.

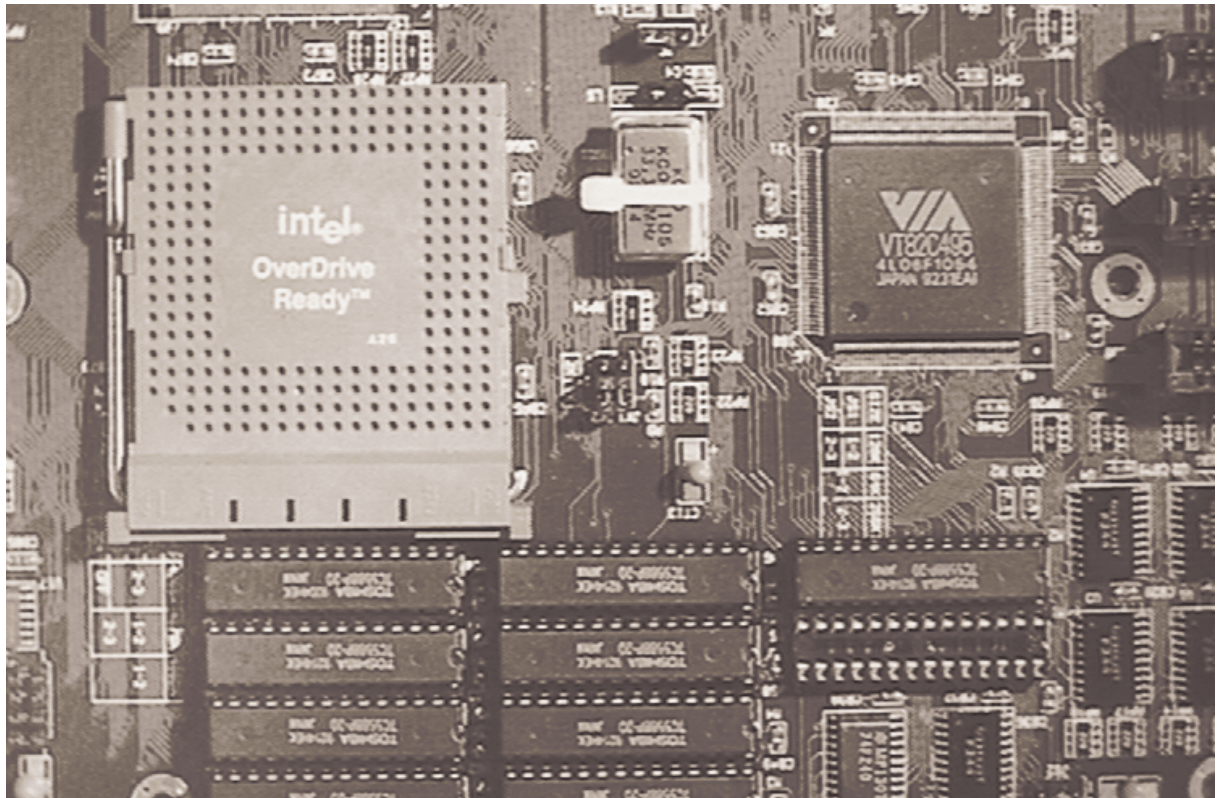


Bild 1: Bildspeicherinhalt der PCgrab



Bild 2: Zu findende Masken

## Eigenschaften des Suchverfahrens

Für die Suche wird ein Verfahren verwendet, daß auf der sogenannten normalisierten Korrelation basiert. Für den mathematisch interessierten Leser, hier die zu Grunde liegende Formel:

$$r(I, M) = \frac{\sum_i I_i M_i - (\sum_i I_i)(\sum_i M_i)}{\sqrt{(\sum_i I_i^2 - (\sum_i I_i)^2)(\sum_i M_i^2 - (\sum_i M_i)^2)}}$$

N bezeichnet die Anzahl der Pixel der Maske

I bezeichnet die Intensität der Pixel im Bild

M bezeichnet die Intensität der Pixel in der Maske

r(I,M) bezeichnet den Korrelationskoeffizienten

Dieses Verfahren kennzeichnet die folgenden Eigenschaften:

- Beleuchtungsunabhängigkeit, d.h. das Verfahren findet die Maske bei beliebigen Beleuchtungsverhältnissen im Bild.
- Rotationsunabhängigkeit, d.h. Rotation der Maske im Bereich von +/- 5 Grad beeinflussen das Suchergebnis nicht.
- Selbst teilweise Verdeckung der Maske im Bild liefert noch ein exaktes Suchergebnis ( im Gegensatz beispielsweise zu Schwerpunktsverfahren).

## Anwendungsbeispiele

Um genügend genaue Suchergebnisse zu erhalten, haben sich in der Praxis folgende Vorgehensweisen bewährt:

### 1) Mittelung der Masken

Bei Masken schlechter Qualität empfiehlt es sich, mehrere Masken zu mitteln und dann zu trainieren. Bild 3 erläutert diesen Sachverhalt an einem Beispiel, wo zwei Masken vorliegen, bei denen Unschärfen am rechten bzw. linken Rand auftreten. Die gemittelte Maske hingegen weist diese Unschärfen nicht mehr auf.

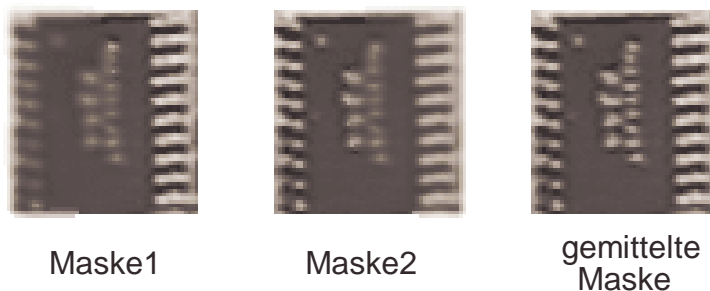


Bild 3: Mittelung der Masken

### 2) Verdrehung der Masken

Wenn in der Anwendung die gelernten Masken in größerer Drehung als +/- 5 Grad auftreten sollten, so sind die Masken beim Trainingsprozess mit unterschiedlichen Drehwinkeln zu trainieren. Die NKF-Bibliothek stellt

natürlich die geforderte Funktionalität zur Verfügung. In Bild 4 wird die grundlegende Vorgehensweise am Beispiel eines SMD-IC's gezeigt. Im vorliegenden Beispiel ist es nun möglich, das SMD-IC unter vier unterschiedlichen Rotationswinkeln zu erkennen. Hierzu wird einfach nicht nur mit der Originalmaske, sondern auch mit den rotierten Masken eine Korrelation durchgeführt. Aus dem Korrelationsmaximum kann nun zusätzlich auch noch die Rotation der Maske bestimmt werden. Unterstützt werden Rotationswinkel von -90 bis 90 Grad.

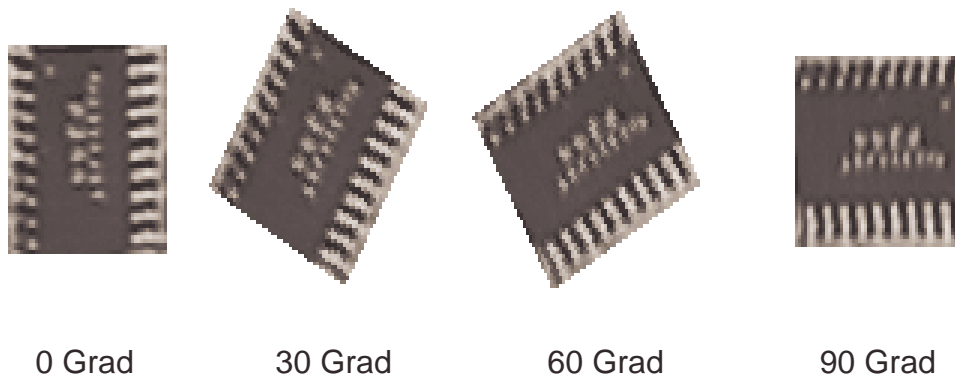


Bild 4: Verdrehung der Masken

### 3) Wahl der *Don't Care* Bereiche

Um gegen leichte Verschiebung und Rotation unempfindliche Masken zu trainieren, bietet die NKF-Bibliothek die sogenannten *Don't Care* Bereiche an, dies sind Bereiche, die keinen Einfluß auf das Korrelationsergebnis haben sollen, wie in Bild 4 gezeigt ( die *Don't Care* Bereiche werden durch schwarze Pixel dargestellt ). Durch geeignete Wahl dieser Bereiche läßt sich überdies hinaus auch Unempfindlichkeit an den Kanten des Objekts erzielen, wenn diese Kanten durch die *Don't Care* Bereiche ausmaskiert werden. Darüberhinaus beschleunigt sich durch die Definition von *Don't Care* Bereichen das Korrelationsverfahren, da weniger Bildpunkte der Maske mit dem gewählten Bildausschnitt korreliert werden müssen.



Originalmaske

Maske mit Don't Care  
Bereichen

Bild 4: Wahl der Don't Care Bereiche

### 4 )Wahl der Graustufen der Maske

Die PCgrab stellt dem Anwender 256 Graustufen zur Verfügung. Anwendungen in der Praxis haben jedoch gezeigt, daß diese Anzahl für das

sichere Erkennen der Masken nicht notwendig sind. Dies hängt jedoch sehr stark vom konkreten Anwendungsfall ab.  
Bewährt haben sich folgende Parameter:

- 16 Graustufen für das Training der Masken
- 24 Graustufen für den gewählten Bildausschnitt

Die Anzahl der Graustufen hat unmittelbaren Einfluß auf die Geschwindigkeit des Korrelationsverfahrens, je niedriger sie ist, desto schneller wird ein Korrelationsergebnis geliefert. Die NKF-Library stellt dem Anwender darüber hinaus parametrisierbare Look-Up-Tabellen zur Verfügung, um eine optimale Graustufenausnutzung im relevanten Bereich zu erzielen. Bild 5 gibt eine Vorstellung von der Auswirkung einer Reduktion der Graustufen von 256 auf 16 Graustufen.



Maske mit 256  
Graustufen



Maske mit 16  
Graustufen

Bild 5: Wahl der Graustufen der Maske

## 5)Wahl des Reduktionsfaktors der Maske

Um die Suchzeiten im Bildausschnitt nach einer Maske ( oder mehreren Masken ) noch weiter zu verkürzen, stellt die NKF-Library einen Reduktionsfaktor für die Maske zur Verfügung, welcher die Bilddaten der Maske um eben dieses Reduktionsfaktor verringert. Im allgemeinen können folgende Reduktionsfaktoren gewählt werden:

- Reduktionsfaktor 3 bei Maskengrößen bis 32x32 Pixel
- Reduktionsfaktor 4 bei Maskengrößen über 32x32 Pixel

Diese Angaben stellen jedoch nur Richtwerte dar, im Einzelfall sollte man so vorgehen, daß zuerst ein Reduktionsfaktor von 0 gewählt wird, und dieser dann schrittweise erhöht wird, solange die Suchergebnisse den gewünschten Anforderungen entsprechen. Bild 6 zeigt im ersten Drittel die schrittweise

Erhöhung von Reduktionsfaktor 0 bis Reduktionsfaktor 3 einer Beispielmaske.



Reduktionsfaktor 0



Reduktionsfaktor 1



Reduktionsfaktor 2



Reduktionsfaktor 3

Bild 6: Wahl des Reduktionsfaktors der Maske

## **NKF-Library für Borland C 3.1**

Im Folgenden werden nun die Funktionen und Datentypen der NKF-Library vorgestellt. Ein auf der PCgrab ablauffähiges Demo-Programm zeigt das Zusammenspiel der einzelnen Funktionen.

**Datentyp: CORR\_MASK\_PARAMS****Aufbau:**

Strukturelement	Datentyp	Beschreibung
x_size	WORD	Größe der Maske in X-Richtung
y_size	WORD	Größe der Maske in Y-Richtung
mask_data	BYTE *	Zeiger auf ein BYTE-Array, daß die Maskeninformation enthält
care_data	BYTE *	Zeiger auf ein BYTE-Array, daß relevante Maskenpunkte von unrelevanten Maskenpunkten spezifiziert (Optional)
look_up_table	BYTE *	Zeiger auf ein BYTE-Array, daß die Look-Up-Table der Maske enthält(Optional)
start	BYTE	Start-Index in der Look-Up-Table
stop	BYTE	Stop-Index in der Look-Up-Table
num_col	BYTE	Anzahl der Graustufen, in denen die Maske intern abgelegt werden soll
reduce_fac	WORD	Reduktionsfaktor, in der die Maske intern abgelegt wird: 0 = keine Reduktion 1 = Reduktion um 2 2 = Reduktion um 4 3 = Reduktion um 8 usw.
rot_angle	FLOAT	Rotationswinkel im Bogenmaß, um dem die Maske gedreht werden soll( $-\pi/2 \leq \text{rot\_angle} \leq \pi/2$ )



**Datentyp: CORR\_MASK****Aufbau:**

Strukturelement	Datentyp	Beschreibung
num_col	BYTE	Anzahl der Graustufen der Maske
reduce_fac	WORD	Reduktionsfaktor der Maske (siehe CORR_MASK_PARAMS)
x_size	WORD	Größe der Maske in X-Richtung
y_size	WORD	Größe der Maske in Y-Richtung
sin_phi	FLOAT	Sinus des Drehwinkels im Bogenmaß
cos_phi	FLOAT	Cosinus des Drehwinkels im Bogenmaß
size	WORD	Größe der Maske in Bytes
mask_data	BYTE *	Zeiger auf die Maskendaten
histo	WORD *	Zeiger auf das Histogramm der Maske
n	DWORD	Anzahl der Pixel der Maske
m	DWORD	Summe der Intensitäten der Maske
m2	DWORD	Summe der Intensitäten zum Quadrat der Maske
a	double	Temporärer Rechenparameter

**Datentyp: SEARCH\_PARAMS****Aufbau:**

Strukturelement	Datentyp	Beschreibung
look_up_table	BYTE *	Zeiger auf ein Byte-Array, das die Look-Up-Table der Bilddaten enthält(Optional)
corr_mask	CORR_MASK *	Zeiger auf eine CORR_MASK Datenstruktur
start	BYTE	Start-Index in der Look-Up-Table
stop	BYTE	Stop-Index in der Look-Up-Table
num_col	BYTE	Anzahl der Graustufen, auf die die Bilddaten zu reduzieren sind
reduce_fac	WORD	Raster, welches für die Korrelation der Bilddaten mit der Maske verwendet werden soll: 0 = jedes Pixel 1 = jedes 2.Pixel 2 = jedes 4.Pixel 3 = jedes 8 Pixel usw.
min_threshold	FLOAT	Spezifiziert eine untere Akzeptanzschwelle für das Korrelationsergebnis
max_count	WORD	Maximale Anzahl der Korrelationsergebnisse, die ermittelt werden sollen
cur_count	WORD	Anzahl der Korrelationsergebnisse, die ermittelt wurden

**Datentyp: SEARCH\_RESULT****Aufbau:**

Strukturelement	Datentyp	Beschreibung
x	WORD	X-Position des Korrelationsergebnis
y	WORD	Y-Position des Korrelationsergebnis
corr	FLOAT	Wert des Korrelationsergebnis
i	DWORD	Summe der Intensitäten an der Stelle des Korrelationsergebnis
i2	DWORD	Summe der Intensitäten zum Quadrat an der Stelle des Korrelationsergebnis
mi	DWORD	Summe der Produkte der Intensitäten der Maske und der Bilddaten an der Stelle des Korrelationsergebnis

## Funktionsname: stdLearnMask

**Syntax:** `BOOL stdLearnMask(CORR_MASK_PARAMS* cmp, CORR_MASK* crm)`

### Beschreibung der Funktion:

Mit **stdLearnMask** wird eine Maske trainiert, die später in einem Bildausschnitt gefunden werden soll.

### Übergabeparameter:

`cmp`: Zeiger auf eine `CORR_MASK_PARAMS` Datenstruktur.  
`crm`: Zeiger auf eine `CORR_MASK` Datenstruktur, **stdLearnMask** belegt diese Struktur.

**Rückgabewert:** `TRUE` bei erfolgreicher Ausführung.  
`FALSE` bei nicht erfolgreicher Ausführung.

### Beispiel:

```
#include "nkf.h"

CORR_MASK_PARAMS crp;
CORR_MASK cm;

BOOL test(WORD w, WORD h, BYTE *data)
{
    crp.x_size = w;
    crp.y_size = h;
    crp.mask_data = data;
    crp.care_data=NULL;
    crp.look_up_table = NULL;
    crp.start = 0;
    crp.stop = 255;
    crp.num_col = 16;
    crp.reduce_fac = 4;
    crp.rot_angle = 0.0;
    return(stdLearnMask(&crp,&cm));
}
```

## Funktionsname: stdLoadMask

**Syntax:** BOOL stdLoadMask(char\* fnm,CORR\_MASK\* crm)

**Beschreibung der Funktion:**

Mit **stdLoadMask** wird eine Maske von Diskette/Festplatte geladen, die später in einem Bildausschnitt gefunden werden soll.

**Übergabeparameter:**

fnm: Zeiger auf einen Dateinamen.  
crm: Zeiger auf eine CORR\_MASK Datenstruktur, **stdLoadMask** belegt diese Struktur.

**Rückgabewert:** TRUE bei erfolgreicher Ausführung.  
FALSE bei nicht erfolgreicher Ausführung.

**Beispiel:**

```
#include "nkf.h"

CORR_MASK cm;

BOOL test(char *fname)
{
    return(stdLoadMask(fname,&cm));
}
```

## Funktionsname: stdWriteMask

**Syntax:** `BOOL stdWriteMask(char* fnm,CORR_MASK* crm)`

**Beschreibung der Funktion:**

Mit **stdWriteMask** wird eine Maske auf Diskette/Festplatte gespeichert, um sie später wieder für das Aufinden in einem Bildaschnitt zu benutzen.

**Übergabeparameter:**

fnm: Zeiger auf einen Dateinamen.  
crm: Zeiger auf eine CORR\_MASK Datenstruktur.

**Rückgabewert:** TRUE bei erfolgreicher Ausführung.  
FALSE bei nicht erfolgreicher Ausführung.

**Beispiel:**

```
#include "nkf.h"

CORR_MASK cm;

BOOL test(char *fname)
{
    return(stdWriteMask(f_name,&cm));
}
```

## Funktionsname: stdFreeMask

**Syntax:** `BOOL stdFreeMask(CORR_MASK* cm)`

**Beschreibung der Funktion:**

Mit **stdFreeMask** gibt den Speicherplatz frei, den `stdLearnMask` und `stdLoadMask` intern benötigen.

**Übergabeparameter:**

`cm`: Zeiger auf eine `CORR_MASK` Datenstruktur.

**Rückgabewert:** `TRUE` bei erfolgreicher Ausführung.  
`FALSE` bei nicht erfolgreicher Ausführung.

**Beispiel:**

```
#include "nkf.h"

CORR_MASK cm;

BOOL test(void)
{
    return(stdFreeMask(&cm));
}
```





