

# Heizungssteuerung durch Geofencing

Fortgeschrittenenpraktikum von Richard Sheljaskow und Yannik Schwarz

## Inhaltsverzeichnis

Zielsetzung.....	1
Beschreibung des Ablaufs.....	1
Dynamische Anpassung eines Fences.....	2
Umsetzung in Code.....	3
Szenarien.....	5
Weit entfernter Arbeitsplatz.....	5
Naher Arbeitsplatz.....	5
Hohe Frequentierung mehrerer Orte.....	5
Verhalten bei mehreren Nutzern.....	5
Fazit.....	5

## Zielsetzung

Das Ziel des Praktikums war es, einen Algorithmus auf Basis von Geofences für eine intelligente Heizungssteuerung zu entwerfen.

## Beschreibung des Ablaufs

Zuerst wird eine geringe Anzahl an festgelegten Standardfences erstellt. Diese haben den Wohnort des Nutzers als Mittelpunkt und haben für alle Nutzer die gleichen festen Radien. Aktuell sind diese 5 Geofences mit einem Radius von 500, 1000, 2000, 5000 und 10000 Meter initialisiert. Jeder dieser Geofences hat eine zugewiesene Temperatur, wonach die Heizung bei Übertritt eines Fences dementsprechend eingestellt wird. So wird die Heizungstemperatur auf Fahrten weg vom Wohnort mit jedem überschrittenen Fence um ein Grad verringert.

Dies ist die grundlegende Funktionsweise dieses Algorithmus und deckt viele normale Nutzungsszenarien ganz automatisch durch seine Struktur ab. Nun gilt es, eine sinnvolle Behandlung für Randfälle zu finden.

Dies wurde durch dynamisches Setzen neuer Geofences realisiert. So werden „Hotspots“ identifiziert, an welchen sich der Nutzer häufig aufhält und viel Zeit verbringt. Man betrachte diese in Zusammenspiel mit den bisherigen Standardfences. Sollten sich diese Hotspots in geringer Entfernung zum Wohnort befinden, so würde sich die Heizung kaum herunterregeln

können, obwohl der Nutzer lange an einem Ort verbleibt und größere Einsparungen möglich wären. Deshalb werden neue Fences gesetzt, welche die Temperatur des nächstkleineren besitzen, um dabei zu helfen, einen solchen „Standby-Zustand“ zu erkennen und auf ihn reagieren zu können. So kann zum Beispiel während des Aufenthalts am Arbeitsplatz die Heizung stark heruntergeregelt werden. Erst beim Verlassen dieses Ortes wird sie wieder auf das normale Niveau der umliegenden Geofences gebracht. Die neuen Fences ermöglichen zudem eine schnelle Erkennung, ob der Nutzer die Arbeit verlässt oder sich nur innerhalb der Arbeitsstätte bewegt. In einem festen Zeitrahmen müssen alle

### Dynamische Anpassung eines Fences

Eine wichtige Überlegung bezieht sich auf die grundsätzliche Idee hinter dem Algorithmus zur dynamischen Anpassung der Geofences. Im Folgenden werden Überlegungen hierzu dargestellt. Man betrachte die Ausgangslage mit einer Handvoll fester Geofences. Nun könnte man das tägliche Nutzerverhalten analysieren und bei einem weit entfernten Arbeitsplatz oder langen Aufenthalten an Orten die Abstände zwischen den Geofences erhöhen und ebenso bei nahegelegenen Arbeitsplatz oder schnellem, irregulärem Bewegungsablauf die Abstände verringern. Macht das Sinn?

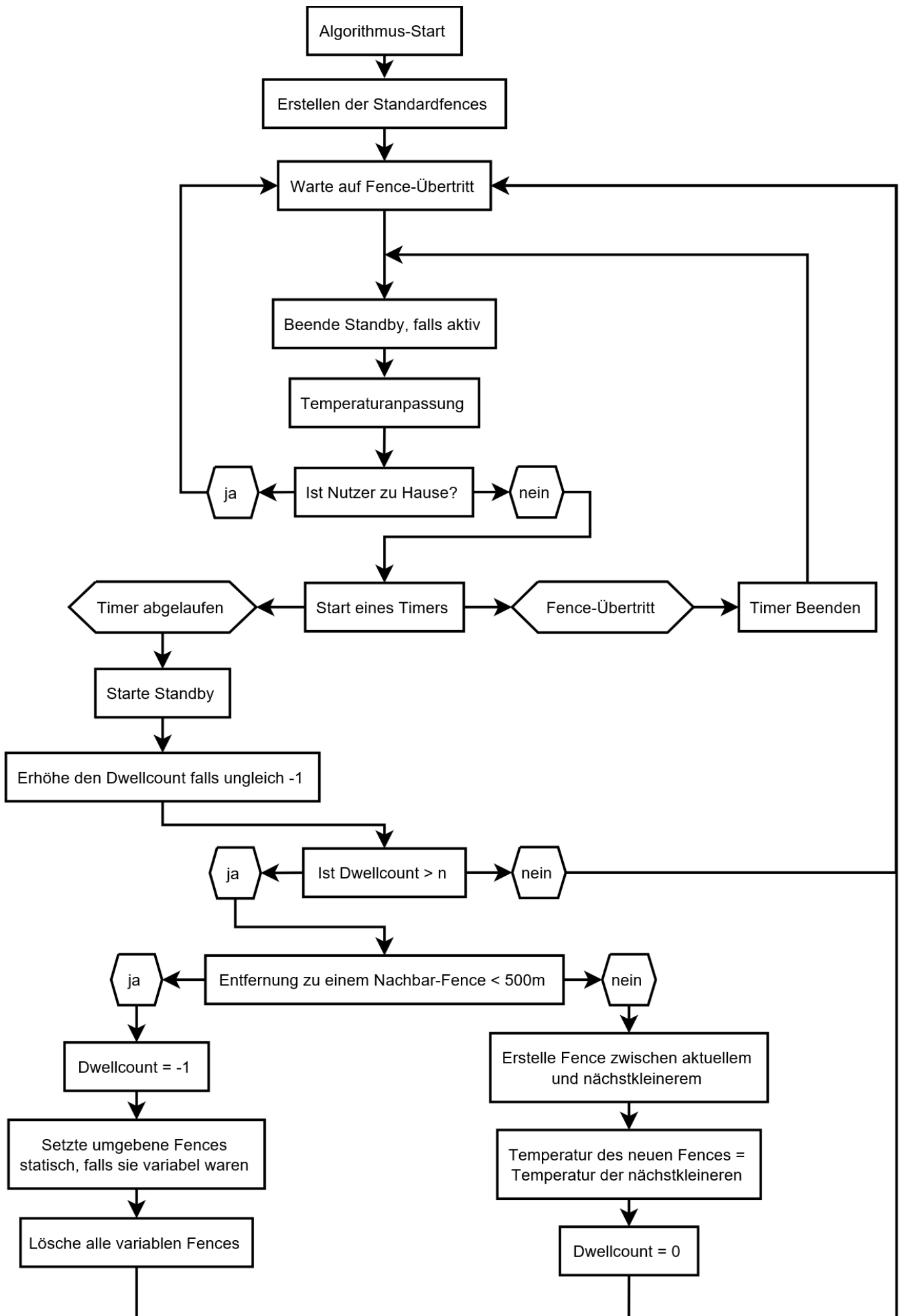
Sei der Arbeitsplatz weit entfernt. Sobald nach dem letzten Fence die Heizung heruntergedrosselt wurde, ist diese auf ihrem niedrigsten Stand. Die Abstände zwischen den Fences zu erhöhen und damit effektiv alle Geofences weiter nach außen zu schieben würde dazu führen, dass die Heizung später heruntergeregelt wird und bei einer Rückkehr nach Hause schon früher wieder hochgeregelt wird. Da die Abstände der Standardfences bereits so gesetzt waren, dass genügend Zeit zur Anpassung verfügbar war, würde dies lediglich zu erhöhtem Energieverbrauch führen. Fährt der Nutzer schneller nach Hause, so erreicht er auch schneller den nächsten Fence und die Heizung bekommt schneller das Signal zum Aufwärmen. Optimierungen in diesem Bereich schienen nicht zielführend. Man betrachte nun den Gegenfall. Werden bei regelmäßigen langen Aufenthalten nahe am Wohnort die Fenceabstände verringert, so können Energieeinsparungen gewonnen werden, da der Arbeitsplatz sich nun in einem niedrigeren Temperaturbereich befindet. Gleichzeitig müsste die Heizung aber auch deutlich schneller aufwärmen und jegliche kleine Einkaufsfahrten nahe des Wohnortes würden ein ungewünschtes herunterdrosseln durch die nun näheren Fences hervorrufen.

Aus diesen Gründen wurde sich gegen eine dynamische Anpassung der gewählten Standardfences und stattdessen für das Ermitteln von Hotspots entschieden.

## Umsetzung in Code

Da in dieser Implementierung lediglich die Information der Geofence-Überschreitungen genutzt wird, müssen auch die Hotspots mit diesen Informationen identifiziert werden. Der im folgenden Absatz beschriebene Ablauf kann auch im beigefügten Diagramm nachvollzogen werden.

Bei Überschreitung eines Fences wird ein Timer gestartet. Falls vor Ablauf ein weiterer Fence überschritten wird, wird der Timer zurückgesetzt und neu gestartet. Falls der Timer ohne weitere Fenceüberschreitungen abläuft, wird der Standby-Zähler des aktuellen Fences erhöht. Dieser Zähler (dwellCount) zählt mit, wie häufig sich der Nutzer über einen längeren Zeitraum in einem Fencebereich aufgehalten hat. Eine Ausnahme bietet der Bereich Rund um den Wohnort, da eine Regelung der Heizung nicht sinnvoll ist, während der Nutzer zuhause ist. Sobald dieser Zähler für einen Fencebereich einen definierter Wert  $n$  überschreitet, wird in der Mitte dieses Bereichs ein neuer temporärer Fence angelegt. Dieser hat noch keine weitere Wirkung, sondern dient zur näheren Bestimmung des Hotspots. Durch vielfache Wiederholung dieses Prozesses wird der Fencebereich immer wieder halbiert und nähert sich langsam an den Hotspot an. Sobald sich der Zähler in einem Bereich erhöht, wo die beiden umliegenden Fences weniger als 500 Meter voneinander entfernt sind, wird dies als hinreichende Annäherung an den Hotspot angesehen. Daraufhin werden die beiden umliegenden Fences statisch gesetzt und ab sofort als normaler Fences behandelt. Alle anderen temporären Fences werden nun gelöscht. Die Zähler aller Fences müssen in einem festen Zeitrahmen von z.B. 30 Tagen zurückgesetzt werden, da sonst falsche Hotspots identifiziert werden würden.



## Szenarien

Im Folgenden werden verschiedene Szenarien vorgestellt, die verschiedene Nutzer darstellen und zeigen, wie der entwickelte Algorithmus damit umgeht.

### Weit entfernter Arbeitsplatz

Hierbei kann zwischen 2 Möglichkeiten unterschieden werden. Ist der Arbeitsplatz außerhalb des letzten Fences, dann wird der Fall eintreten, dass der Timer abläuft und die Heizung in den Standby-Modus wechselt. Ist der Nutzer vor dem letzten Fence, wird pro Ablauf des Timers der Zähler solange erhöht bis nach einigen Tagen ein neuer Fence in der Mitte des betrachteten Fencebereichs erstellt worden ist. Dabei wird dies solange fortgesetzt, bis die Fences sich einander um 500m angenähert haben. Dadurch kann nun schnell herausgefunden werden, wenn der Nutzer seinen Arbeitsplatz verlässt.

### Naher Arbeitsplatz

Ein naher Arbeitsplatz verhält sich genauso wie ein weit entfernter Arbeitsplatz. Die Heizung wird jedes Mal erfolgreich durch Ablaufen des Timers in den Standby-Zustand gebracht. Dabei ist jedoch zu beachten, dass aufgrund der Nähe die Heizung nicht auf optimaler Temperatur bei Erreichen der Wohnung sein könnte.

### Hohe Frequentierung mehrerer Orte

Während der Nutzer z.B. regelmäßig einer Vereinstätigkeit nachgeht, muss dafür gesorgt werden, dass die Temperatur angepasst wird. Das wird einerseits durch den Timer abgefangen, indem automatisch in den Standby gegangen wird. Andererseits wird sich mit der Zeit ein neuer Fence bilden, wodurch präziser erkannt werden kann, ob der Nutzer zur Zeit an einem bestimmten Ort ist.

## Verhalten bei mehreren Nutzern

Für mehr als einen Nutzer könnte ein deutlich komplexeres System entwickelt werden. Zuerst sollte ein Grundverhalten eingeführt werden, dass nur der Nutzer mit der höchsten Temperatur bzw. der kürzesten Entfernung zum Wohnort die Heizungstemperatur kontrollieren. Schließlich sollte es niemand kalt haben, während man zuhause ist.

Eines der Hauptprobleme mit mehreren Nutzern dürfte sein, dass die Heizung direkt nach einem herab-drosseln, wieder aufgewärmt werden soll. Bei stark unterschiedlichen Tagesroutinen könnte dabei sehr viel Energie verloren gehen. Dies könnte verbessert werden, indem die täglichen Abläufe genauer analysiert und gespeichert werden, um daraus ein prognostiziertes Bewegungsmuster aufzubauen. Verlässt das letzte Familienmitglied das Haus, aber die App sagt die baldige Ankunft eines anderen Familienmitglieds hervor, so würde die Heizung nicht gedrosselt werden und wäre sofort bereit ohne Energie auf rasche Temperaturänderungen zu verschwenden.